A Conditional Random Field Framework for Robust and Scalable Audio-to-Score Matching

Cyril Joder, Slim Essid and Gaël Richard, *Senior Member, IEEE*

***Abstract*—In the present work, we introduce the use of Conditional Random Fields (CRFs) for the audio-to-score alignment task. This framework encompasses the statistical models which are used in the literature and allows for more flexible dependency structures. In particular, it allows observation functions to be computed from several analysis frames.**

Three different CRF models are proposed for our task, for different choices of tradeoff between accuracy and complexity. Three types of features are used, characterizing the local harmony, note attacks and tempo.

We also propose a novel hierarchical approach, which takes advantage of the score structure for an approximate decoding of the statistical model. This strategy reduces the complexity, yielding a better overall efficiency than the classic beam search method used in HMM-based models.

Experiments run on a large database of classical piano and popular music exhibit very accurate alignments. Indeed, with the best performing system, more than 95% of the note onsets are detected with a precision finer than 100 ms. We additionally show how the proposed framework can be modified in order to be robust to possible structural differences between the score and the musical performance.

I. Introduction

We address the problem of audio-to-score alignment (or audio-to-score matching), which is the task of synchronizing an audio recording of a musical piece with the corresponding symbolic score. Depending on the targeted application, the alignment can be either *on-line* or *off-line*.

On-line alignment has been extensively considered in the context of *score following*, that is the real-time tracking of a musician performance (see for example [1], [2], [3]). Such a tracking allows for an automated computer accompaniment of a live soloist, but also permits other kinds of interactions between human performers and automated processes. This may include real-time sound transformations (e.g. auto-tuning) or the control of visual effects which are to be synchronized with the music (e.g. stage lights or opera supertitles).

In off-line audio-to-score matching, the whole performance is accessible for the alignment process. This permits to develop non causal algorithms that can reach higher matching precision. This can be interesting for applications that do not require the real-time property. Among the most immediate perspectives is, for instance, the possibility to browse a musical recording using the symbolic score [4]. An efficient audio-to-score alignment also opens the path for new or emerging applications in the field of Music Information Retrieval (MIR) such as MIDI-informed audio indexing, transcription or source separation [5], by taking advantage of the numerous scores that can be found on the Internet.

The requirement for an ideal audio-to-score alignment system may be manifold. In particular, they should comprise generality, precision, scalability and robustness. Generality refers to the capacity of the system to handle a wide variety of musical styles. Yet many works on this task are limited, for example to monophonic music [6], [7] or to single-instrument music [8], [9]. Moreover, most international evaluation campaigns have used data from a single style with low polyphony (single instrument classical music) and this limitation was only addressed in the last MIREX evaluation campaign[1].

Scalability refers to the ability of a system to define different operating points with varying precision/complexity tradeoff. It should be an important property of an alignment system, which, to the authors' knowledge, has rarely been considered in previous works. The desired tradeoff between precision and complexity of the alignment process may depend on the target application. For instance, low complexity would be favored for retrieval applications whereas high precision would be preferred for (off-line) informed source separation tasks.

Robustness refers in particular to the ability of the system to deal with imperfect or low quality scores (or even low detail scores, such as lead sheets). The variation of the musical structure is certainly among the main sources of difference between an audio recording and its corresponding score (e.g. omission of a repeat section in classical music, additional solo in a live performance in pop music, . . .).

Audio-to-score matching is traditionally performed in two steps, namely *feature extraction* and *alignment*. The features extracted from the audio signal characterize some specific information about the musical content. Then the alignment is performed by finding the best match between the feature sequence and the score.

The features used in audio-to-score alignment or score following systems always comprise a descriptor of the instantaneous pitched content of the signal. Early works exploited the output of a fundamental frequency detector [6], [10]. But such estimators are error-prone, especially in the context of polyphonic music. That is why many systems

This work has been partly supported by the Quaero Program, funded by OSEO, French State agency for innovation.

The authors are with the Institut TELECOM, TELECOM ParisTech, CNRS LTCI, F-75634 Paris, FRANCE (e-mail: cyril.joder@telecom-paristech.fr, slim.essid@telecom-paristech.fr, gael.richard@telecom-paristech.fr)

[1] Music Information Retrieval Evaluation eXchange 2010, score following task: http://www.music-ir.org/mirex/wiki/2010:Real-time_Audio_to_Score_Alignment_(a.k.a_Score_Following)

employ low-level features such as the output of a Short-Time Fourier Transform (STFT) [11], the energy in frequency bands corresponding to the musical scale [12] or *chromagram* features. The *chromagram* representation, which collapses the previous energy bands into one octave, is extensively used in off-line audio-to-audio and audio-to-score alignment [13], [14]. This representation has the advantage of being robust to octave errors, as well as to timbral changes. The information conveyed by note attacks may also be useful to precisely detect the note onsets. Thus, some existing works take advantage of onset features, such as the derivative of the short-time energy [6], [7]. In [15] a "chroma onset" feature is designed, which encompasses the onset information in each chroma bin.

Regarding the alignment process, early works on real-time score following, as well as most off-line systems (for example [15], [16]) rely on cost measures between events in the score and in the performance. The alignment is then obtained by "warping" the score so as to minimize the cumulative cost, using dynamic programing techniques. The Dynamic Time Warping (DTW) algorithm, or variants thereof, are extensively used [15], [17]. This approach has the advantage of being computationally simple and can be applied to audio-to-audio synchronization. A hierarchical version of DTW [18] has been applied to this task, either for complexity reduction [19] or accuracy refinement [20]. Robustness issues, for example to structure changes, have been tackled by allowing partial synchronization path searches in the DTW alignment [21] or in the online context, by running several *trackers* in parallel [22].

In the recent literature, score following systems have employed probabilistic models such as Hidden Markov Models (HMMs) [23] in order to take into account the uncertainty of the matching [9], [12], [24]. In such systems, the hidden variable represents the current position in the score. Flexible transition probabilities can also permit possible structure changes. However as far as we know, the applicaton of probabilistic models to this problem has only been tackled in [24]. More sophisticated models handling a continuous variable for position in the score [25], or a tempo variable [11], [26], [27] have been proposed. In these latter models, the note duration probabilities can be explicitly modeled as functions of the current tempo.

The previous statistical models belong to the class of Bayesian Network (BN) models [28]. BNs are used for alignment as generative models, which suppose a prior distribution of the hidden variables and the probabilities of the observations given these hidden variables. However in our alignment problem, the goal is to find the values of the hidden variables, given the observations. Thus, a *discriminative* framework can be employed. In this work, we introduce the use of Conditional Random Fields (CRFs) [29], for the purpose of audio-to-score alignment. These models have scarcely been applied to the field of Music Information Retrieval (MIR) and, to the authors' knowledge, never been used for alignment. Yet, CRFs represent a number of advantages over BNs. First, in the context of alignment, they can be seen as a generalization of BN. Second, the CRF framework enables the relaxation of some conditional independence assumptions of BNs, thus authorizing more general dependency structures, without increasing the decoding complexity.

The contributions of the present work are multifold.

- We reformulate the audio-to-score alignment in the context of CRF and show that the models in the literature can be presented as CRFs with possibly different structures.
- We show that CRFs are particularly well suited to design flexible observation functions. In particular, we consider a whole neighborhood of a time frame, for a more precise matching of this frame with a score position.
- We show how different types of features can be efficiently exploited inside this framework. Namely, chroma features characterizing the instantaneous harmony of the signal, spectral flux detecting the note onsets and the "cyclic tempogram" feature [30] characterizing the local tempo.
- Additionally, we propose a novel hierarchical decoding approach, which takes advantage of a hierarchical segmentation of the music into *concurrencies* (units of constant-pitch content), *beats* and *bars*. This method allows for a scalable alignment since it can be used to reduce the search space in the decoding process, or to obtain a fast, coarse alignment.
- We conduct an extensive experimental study on polyphonic, multi-instrumental music, over a database of significant size and test the robustness and scalability of the proposed framework.

The rest of this paper is organized as follows. The graphical model framework is presented in Section II. The transition models and the observation models of the CRF models are respectively detailed in Sections III and IV. Then, a hierarchical pruning method for an approximate decoding of these models is proposed in Section V and the experimental results are exposed in Section VI, before suggesting some conclusions.

II. Conditional Random Fields for Alignment

In this section, we introduce the use of Conditional Random Fields (CRFs) as a unified framework for the probabilistic models used in the audio-to-score alignment task. We first formalize our alignment task as a sequence labelling problem, before presenting CRFs. We explain how this formalism can represent the statistical models of the literature, and show that it also enables new modeling possibilities.

A. Alignment as a Sequence Labelling Problem

Following [31], we define a *concurrency* as a set of notes that sound at the same time. The musical score can be represented as a sequence of notes (in monophonic music) or concurrencies in the case of polyphonic music, as illustrated in Fig. 1. In order to deal with recordings starting before the first note of the score or stopping after the end of the last note, we create artificial empty concurrencies (labeled with numbers 0 and 7 in the figure) at the beginning and the end of the concurrency sequence.

The goal of audio-to-score alignment is then to find in the musical recording the positions of the concurrencies given; or equivalently, to find in the score the position corresponding to each time frame of the performance. This can be seen as a

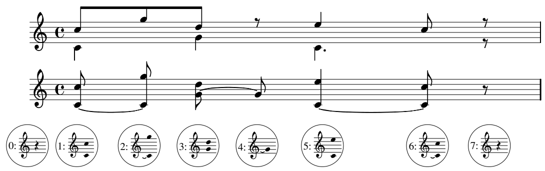


Figure 1. Representation of a score as a sequence of concurrency labels. The original score (up) is segmented into units of homogeneous polyphony: each note onset or offset corresponds to a new concurrency. Middle: the concurrency sequence representation. Bottom: the concurrency labels.

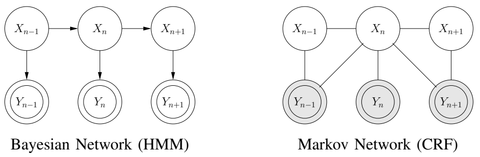


Figure 2. Comparison between HMM and CRF graphical representations. Double nodes represent observed variables and shaded nodes (in CRF) correspond to variables that the model conditions on.

labelling problem, whose labels are possible score positions, for example the score concurrencies.

The audio recording is converted into a discrete sequence of *observation features*, which describe the instantaneous content of the signal over short frames. Let $\mathbf{y}_{1:N} = y_1 \dots y_N$ be this observation sequence, of length N. Our task is to find the label sequence that best matches the audio data.

The use of a probabilistic model allows for the calculation of the optimal label sequence given the observation sequence. Let $\mathbf{X}_{1:N} = X_1, \dots, X_N$ be an (unobserved) random process representing the labels. As in [27], we use the Maximum *a Posteriori* (MAP) criterion, which defines the optimal label sequence $\hat{\mathbf{x}}_{1:N}$ as:

$$\hat{\mathbf{x}}_{1:N} = \underset{\mathbf{x}_{1:N}}{\operatorname{argmax}}\, p\left(\mathbf{X}_{1:N} = \mathbf{x}_{1:N} | \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\right). \qquad (1)$$

For the sake of clarity, the boundary indices ${}_{1:N}$ will be omitted in the following when no ambiguity is introduced.

B. Conditional Random Fields (CRFs)

A Conditional Random Field (CRF) is a type of *discriminative* undirected graphical model (see [32] for an introduction on graphical models) initially introduced for the labelling or segmentation of sequential data [29].

The graphical representation of a CRF is thus an undirected graph, as depicted in Fig. 2 (right). Whereas a *generative* model (such as an HMM) specifies the marginal distribution of the hidden variable $p\left(\mathbf{X}\right)$ and the conditional distribution of the observations given the hidden variables $p\left(\mathbf{Y}|\mathbf{X}\right)$, a discriminative model such as a CRF conditions the probabilities on the observation sequence. Hence, it directly models the conditional distribution $p\left(\mathbf{X}|\mathbf{Y}\right)$ which is used in our optimization problem of (1).

As in a HMM, the label process $\mathbf{X}$ is assumed to be Markovian. Thus, the conditional probability of (1) can be factorized as:

$$p(\mathbf{X}|\mathbf{Y}) = \frac{1}{Z}\phi(X_1, \mathbf{Y}) \prod_{n=2}^{N} \psi(X_n, X_{n-1}, \mathbf{Y})\phi(X_n, \mathbf{Y}) \qquad (2)$$

where ψ and ϕ are non-negative *potential functions*. The *transition function* ψ controls the transitions between the labels and the *observation function* ϕ links the current label with the observations. Z is a normalizing factor. Note that in the general form of CRFs, the potential functions can depend on the time frame n. However, we use a simpler model with constant ϕ and ψ functions, since this dependency is difficult to interpret in our case. An exception is made for the first and last observation functions, which can express some constraints at the extremities of the sequence. For example, one can set $\phi(X_1, \mathbf{Y}) = \mathbf{1}_{\{X_1 \leq 1\}}$ (where $\mathbf{1}$ is the indicator function) if one knows that the recording starts with the first concurrencies of the score ($X_1 = 0$ or $X_1 = 1$).

One of the main advantages of CRFs is the relaxation of a conditional independence assumption of HMMs. In an HMM, an observation Y_n is supposed to be independent of all the other variables, given the hidden variable X_n. Thus, no direct dependency can be modeled between a hidden variable and a "remote" observation. In a CRF, no assumption is made on the observation process. Hence, the whole feature sequence can be used in the calculation of the observation function, without increasing the decoding complexity. Indeed, the most probable label sequence of (1) can be calculated by the Viterbi algorithm with the same complexity as in a HMM. This property allows for using a whole neighborhood of each observation as a clue for the labelling of the corresponding time frame. For further details on CRFs, see [29], [33].

It is worth noting that, for a labelling task, a HMM can be seen as a particular case of a CRF. For a given HMM, we construct an undirected graph with the same links as the directed graph of the HMM. Then, we set, for $n \geq 2$:

$$\begin{aligned} \psi(X_n, X_{n-1}, \mathbf{Y}) &= \psi(X_n, X_{n-1}) &&= p_{\mathcal{H}}(X_n|X_{n-1}) && (3)\\ \phi(X_n, \mathbf{Y}) &= \phi(X_n, Y_n) &&= p_{\mathcal{H}}(Y_n|X_n) && (4) \end{aligned}$$

where $p_{\mathcal{H}}$ denotes the probability according to the given HMM. The special case is $\phi(X_1, \mathbf{Y}) = p_{\mathcal{H}}(Y_1, X_1)$. In that case, the conditional probability of (2) can be written

$$p\left(\mathbf{X}|\mathbf{Y}\right) \propto p_{\mathcal{H}}(Y_1, X_1) \prod_{n=2}^{N} p_{\mathcal{H}}(X_n|X_{n-1}) p_{\mathcal{H}}(Y_n|X_n). \qquad (5)$$

Hence, from a decoding point of view, the obtained CRF is equivalent to the given HMM. Since the CRF framework encompasses many different models, including HMMs, we will use the CRF formalism in the rest of this paper to present different alignment models in a unified framework.

III. Choice of Transition Functions

Several types of graphical model structures have been used in previous works on alignment, in order to model the musical performance. This corresponds to different choices for the transition function ψ, which controls the prior probabilities of the label sequences. Since in the alignment task the concurrency succession is given by the score, the purpose of the transition functions is an accurate modeling of the concurrency durations. In this section, we propose a unified viewpoint of

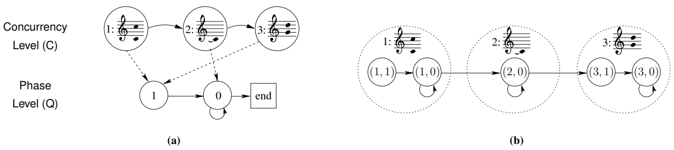


Figure 3. **(a)**: Label automaton of a basic hierarchical structure. In this example, the sub-label 1 corresponds to the *attack* phase whereas the sub-label 0 denotes no attack. **(b)**: The corresponding “flattened” automaton.

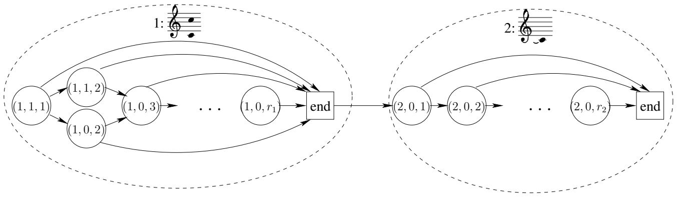


Figure 4. Sub-label automata of two concurrencies. The first one contains an onset, the second on does not. The three variables represent respectively the concurrency C_n, the attack indicator A_n and the occupancy D_n.

the graphical models used in the alignment literature, and transpose them into the CRF framework by exhibiting the corresponding transition functions.

A. Markovian CRF (MCRF)

1) Markovian Transition Function: The Markovian CRF (MCRF) is the transposition of the HMM into the CRF context. In this case, the label sequence is the sequence $\mathbf{C}_{1:N}$ of the played concurrencies. Note that the musical score provides not only the set of concurrencies of the piece, but also the order in which they are to be played. Thus, if this order is trusted, there are only two possible transitions from one concurrency: to the same one or to the following one. The transition function is then of the form:

$$\psi(C_{n+1}, C_n) = \begin{cases} \lambda_0(C_n) & \text{if } C_{n+1} = C_n \\ \lambda_1(C_n) & \text{if } C_{n+1} = C_n + 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

As in a HMM, the prior probability of a concurrency length L can be written $P(L = l) \propto \lambda_0^{l-1}\lambda_1$. Thus, if an anticipated length l is assumed for the concurrency c, the parameters can be set to the values of the HMM transition probabilities which maximize the probability of the duration l. These values are $\lambda_0 = \frac{l-1}{l}$ and $\lambda_1 = \frac{1}{l}$. In the case where no prior duration is known, both parameters are set to 1 so that all possible durations have the same probability.

2) Hierarchical Label Structure: One of the limitations of the previous structure is that the observations extracted from the signal are supposed to be constant during a whole concurrency duration. Indeed, it cannot take into account the variations that may occur inside a concurrency.

However, modeling different phases within a concurrency (for example *attack* and *sustain* phases) can be handled with a structure equivalent to Hierarchical Hidden Markov Models [34]. Here, a lower level of labels is used, representing the phase. For each frame n, let A_n be a binary random variable, which is equal to 1 iff the frame n corresponds to an *attack* phase. The labels of the model are then: $X_n = (C_n, A_n)$.

Fig. 3 displays an example ofautomaton representation for a hierarchical structure. Once a new concurrency is reached, the system enters a sub-label thanks to a unique “vertical transition” (in dotted line). In the lower level of hierarchy, it follows “horizontal transitions” (solid lines) until it reaches an *end* state. Then, it switches back to the higher level and follows a horizontal transition in this level.

A hierarchical structure can in fact always be converted to a “flat” automaton [28], as represented in Fig. 3 (b). However, the hierarchical structure allows for a more intuitive interpretation of the labels.

For concurrencies which do not contain an onset *i.e.* corresponding to note extinctions (such as the second concurrency of Fig. 3), there is no *attack* phase. For the other concurrencies, the first sub-label always corresponds to an *attack*. For a concurrency c, let $\Omega(c)$ be a binary function which indicates if c contains an onset. The transition function is then

$$\psi(X_{n+1}, X_n) = \begin{cases} \lambda_0(C_n) & \text{if } C_{n+1} = C_n \text{ and } A_{n+1} \leq A_n \\ \lambda_1(C_n) & \text{if } \big(C_{n+1}, A_{n+1}\big) = \big(C_n + 1, \Omega(C_{n+1})\big) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The constraints which are added here express that an attack phase ($A = 1$) cannot occur *after* a sustain phase ($A = 0$) of the same concurrency, and that the first frame of an “onset concurrency” corresponds to an attack phase.

B. Semi-Markov CRF (SMCRF)

The main drawback of the Markovian models presented so far is that they cannot favor a particular concurrency length. In order to introduce flexible duration priors, one can exploit semi-Markov models, in which the label durations are explicitly modeled. In our CRF context, we will call such a model a Semi-Markov CRF (SMCRF).

A semi-Markov model can be represented as a hierarchical label automaton, in which sub-labels are introduced for the temporal modeling. The prior duration distribution depends on the number of these sub-labels and the values of their transition functions. Some possible structures are compiled in [35]. Many of the works on alignment exploiting statistical models use similar models (for example [7], [12], [9]).

The sub-label structure used is depicted in Fig. 4. We introduce a variable D_n representing the concurrency *occupancy* *i.e.* the time elapsed since the beginning of the current concurrency. The durations are modeled by letting the transition function depend on this occupancy variable. Note that the number of occupancy sub-labels r_c is the maximum duration of a concurrency c. Furthermore, we constrain the *attack* phase of a concurrency to last one or two frames, as represented in Fig. 4 (first concurrency). Thus, all combinations of variables $X_n = (C_n, A_n, D_n)$ are not possible. In all the following equations, we will suppose that the given label values are admissible. The transition function is then

$$\psi(X_{n+1}, X_n) = \begin{cases} 1 & \text{if } (C_{n+1}, D_{n+1}) = (C_n, D_n + 1) \\ p(L_{C_n} = D_n) & \text{if } (C_{n+1}, D_{n+1}) = (C_n + 1, 1) \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where L_{C_n} denotes the random variable representing the duration of the concurrency C_n.

C. Hidden Tempo CRF (HTCRF)

A limitation of semi-Markov models appears for the temporal modeling of music. Indeed, such models use a unique duration distribution, whereas in a musical performance, the concurrency durations depend on the instantaneous tempo. Hence, the most elaborate models for audio-to-score matching [27], [11] use yet another variable which models the tempo process. Let T_n be this tempo variable. The value of T_n is the tempo, given in number of frames per beat. The possible labels are then of the form $X_n = (C_n, A_n, D_n, T_n)$. We call such a model a *Hidden Tempo CRF* (HTCRF).

In this model, the concurrency duration distribution depends on the current value of the hidden variable T. For example in [27], Raphael models the conditional duration distribution of a concurrency c, given the tempo $T = t$ by a normal distribution centered on the value $\ell_c t$, where ℓ_c is the theoretical length (in beats) of c, with a variance set as a parameter. Our model is similar, but the used variance is proportional to the expected concurrency length (given the tempo). This choice is supported by psychoacoustic results, since the just-noticeable difference reported in [36] is proportional to the original length, for lengths between 240 ms and 1 s. Let $\ell_c t$ be the expected duration of a concurrency. If the actual length of this concurrency is l, we set the duration deviation penalty $h_\mathrm{d}(l, \ell_c t)$ as:

$$h_\mathrm{d}(l, \ell_c t) = \exp\left(-\gamma_\mathrm{d} \left| \frac{l - \ell_c t}{\ell_c t} \right|^2 \right) \tag{9}$$

where γ_d is a parameter controlling the deviation tolerance.

For an intuitive interpretation of the tempo variable, the value of this variable is forced to stay constant over a whole concurrency duration, and changes only occur at concurrency transitions. Following [37], we assume that the tempo changes are relative rather than absolute and that for example, the probability is the same for doubling the tempo and for halving it. Thus, the tempo variation penalty h_t is set to

$$h_\mathrm{t}(t_{n+1}, t_n) = \exp\left(-\gamma_\mathrm{t} \left| \log \frac{t_{n+1}}{t_n} \right|^2 \right), \tag{10}$$

controlled by the parameter $\gamma_\mathrm{t} \geq 0$. When the ratio between two tempi is greater than 2 (or less than $1/2$), we consider that an abrupt tempo change occurs, in which all the possible tempi have the same probability. Thus, we limit the term $\left|\log \frac{t2}{t1}\right|$ to $\log 2$. Finally, the global transition function is

$$\psi(X_{n+1}, X_n) = \begin{cases} 1 & \text{if } (C_{n+1}, D_{n+1}, T_{n+1}) = (C_n, D_n + 1, T_n) \\ h(X_{n+1}, X_n) & \text{if } (C_{n+1}, D_{n+1}) = (C_n + 1, 1) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

with $h\left(X_{n+1}, X_n\right) = h_\mathrm{d}(D_n, \ell_{C_n} T_n)\, h_\mathrm{t}(T_{n+1}, T_n)$. The first term of (11) expresses the case where no concurrency change occurs. In this case, the final duration of the unfinished concurrency is unknown and the tempo stays the same. Thus, no penalty is applied. The second term corresponds to the case where the current concurrency ends. The concurrency duration is then equal to the occupancy D_n, and the corresponding penalty is calculated, as well as the tempo change penalty.

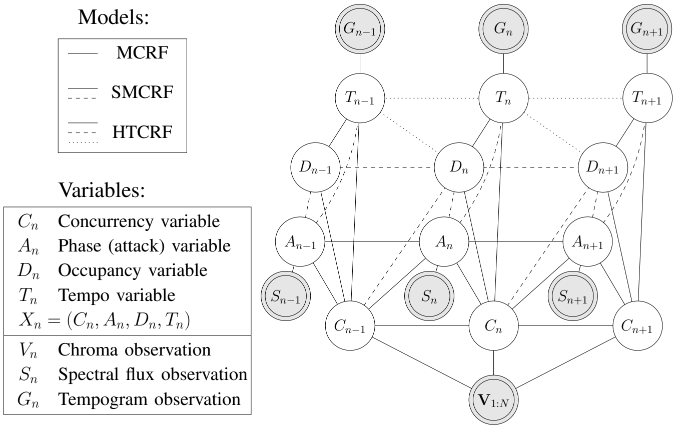


Figure 5. Graphical model representation of the presented models. The solid lines represent Markovian CRF links. Dashed arcs and dotted arcs are additional links of Semi-Markov CRF and Hidden Tempo CRF respectively. The observations are presented in the next section.

The dependency structure of the presented models are represented in Fig. 5. Note that although the occupancy and tempo variables D_n and T_n do not intervene in the MCRF transition function, they do appear in the transition function, as will be detailed in Section IV. The same holds for the tempo variable in the SMCRF model.

IV. Observation Functions

As presented in II-B, the observation function links the current label with the observations. The observations should then reflect the information represented by the labels. We use, for each frame n, three types of features which will be presented in the next subsections: a *chroma* feature v_n, an *onset* feature f_n and a *tempo* feature g_n. The global observation vectors can then be written $y_n = (v_n, f_n, g_n)$. We recall that the possible labels have the form $x_n = (c_n, d_n, a_n, t_n)$, where the variables stand respectively for the current concurrency, occupancy (time elapsed since the beginning of the concurrency), phase and tempo.

Note that, although the occupancy and phase variables are highly correlated, they are not completely redundant. Indeed, since the attack phase of an “onset concurrency” can last one or two frames, the second frame, corresponding to the occupancy $D = 2$, can be either in attack ($A = 1$) or sustain phase ($A = 0$). This is represented in the first concurrency of Figure 4.

We assume that the observation function $\phi(x_n, \mathbf{y}_{1:N})$ has the form:

$$\phi(x_n, \mathbf{y}_{1:N}) = \phi_\mathrm{c}\left(x_n, \mathbf{v}_{1:N}\right) \phi_\mathrm{a}(a_n, \mathbf{f}_{1:N}) \phi_\mathrm{t}(t_n, \mathbf{g}_{1:N}). \tag{12}$$

A. Chroma vectors

Chroma vectors are extracted according to [38], with a time resolution of 20 ms, and normalized so that they sum to 1. For each concurrency label, we build a synthetic chroma vector template from the content of this concurrency, as in [13], [39]. The template is normalized so that it can be regarded as a probability distribution over the chroma bins.

Under the assumption that the tempo can be considered as constant over short time windows, the knowledge of the score position and the current tempo at a time frame is sufficient

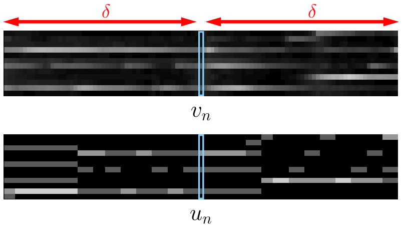


Figure 6. Calculation of the chroma observation function. Up: observations around frame n; Down: chroma templates created as the rendition of the score around score position (c_n, d_n), at constant tempo t_n (which is faster than the interpretation tempo in this example).

to determine the score positions corresponding to any frame in the given window. Then, the corresponding concurrency sequence can be compared to the observation sequence, in order to obtain a ”matching measure” between the observations and the position/tempo hypothesis.

Formally, let $2\delta+1$ be the number of frames over which the tempo is considered as constant (in practice, we set $\delta = 50$, for a length of 2 s). Given a label $x_n = (c_n, d_n, a_n, t_n)$, let $\vec{\mathbf{c}}(x_n)$ be the sequence of concurrency labels corresponding to the theoretical rendition of the score at the constant tempo t_n around (c_n, d_n) –such that the score position at time n is (c_n, d_n). This sequence is illustrated in Fig.6. Let $u_1, \ldots, u_N$ be the corresponding chroma templates. The observation function $\phi_c(x_n, \mathbf{v})$ associated to the chroma vector sequence is:

$$\phi_{\mathrm{c}}(x_n, \mathbf{v}) = \exp - \sum_{k=-\delta}^{\delta} \mu_k D_{\mathrm{KL}}\left(v_{n+k} \middle\| u_{n+k}\right) \tag{13}$$

where μ_k are parameters controlling the weights given to the chroma observations at the different time-shifts. $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence. Intuitively, the weights μ_k should be decreasing with $|k|$, so as to emphasize the current observation v_n. We use an exponential window with decaying parameter 100. The weights are normalized so that the sum is equal to a parameter α.

Note that in the case $\delta = 0$, the value of $\phi_{\mathrm{c}}(x_n, v_n)$ does not depend on the tempo and duration variables.

B. Onset Feature

In order to discriminate the *attack* phase from the *sustain* phase of a concurrency, a straightforward yet efficient onset detector function based on spectral flux is used [40]. The spectral flux is calculated on 40-ms windows with a 20-ms hop-size. A local threshold is computed by applying a 67% rank filter of length 200 ms to the spectral flux values. We then obtain our “onset feature” by subtracting this value to the spectral flux. Finally, a simple logistic model is applied and the observation function ϕ_{a} associated to this onset feature is:

$$\phi_{\mathrm{a}}(a_n, \mathbf{f}) = \mathbf{1}_{\{a_n=0\}} + \mathbf{1}_{\{a_n=1\}} \exp\left(\mu_{\mathrm{f}} f_n\right) \tag{14}$$

with the positive parameter μ_{f}.

C. Cyclic Tempogram

As a feature accounting for tempo, we adopt the *cyclic tempogram* representation [30], which provides a mid-level representation of the tempo. The local autocorrelation of the *spectral flux* is first computed over sliding 5-s windows, for time-lags between $\tau_{\min} = 200$ ms and $\tau_{\max} = 3.2$ s. Let $w_n(\tau)$ be the value of this autocorrelation function for a window centered on frame n.

Similarly to a *chromagram*, the time lags are separated into *octave equivalence classes*: two time-lags τ_1 and τ_2 are octave equivalent iff there is a $k \in \mathbb{Z}$ s.t. $\tau_1 = 2^k \tau_2$. The value $g_n(\tau)$ of the cyclic tempogram for a time-lag τ is calculated by adding all the values of this autocorrelation function corresponding to the same equivalence class:

$$g_n(\tau) = \sum_{k\in\mathbb{Z}} w_n(2^k \tau). \tag{15}$$

In practice, this sum is limited to the time-lags which are considered in the previous step. The value of the observation function ϕ_{t} associated to this tempo feature is then: $\phi_{\mathrm{t}}(y_n, \mathbf{g}) = \exp\left(\mu_{\mathrm{t}} g_n(t_n)\right)$, where μ_{t} is a positive parameter.

V. A Hierarchical Pruning Approach for Approximate Decoding

A. A Hierarchical Model

As presented in Section II, the optimal label sequence $\hat{\mathbf{y}}$ defined in (1) can be found thanks to the Viterbi algorithm, if the label set is finite. However, the complexity of this algorithm is quadratic in the number of possible labels, which can be very large in intricate models such as HTCRF, where the label set is the set of all possible combinations of the hidden variable values. Thus, the cost of a full-fledged Viterbi decoding can be important and pruning methods are often used in order to reduce the explored label set at each iteration.

The usual strategy for HMM and Bayesian Networks in general (for example [27]) is *beam search*, which consists in maintaining only the “most promising” partial label sequences during decoding. If a fixed (small) number of hypotheses is explored at each step, the complexity becomes linear in the number of labels. The partial label sequences are usually ordered according to conditional probability $\bar{p}_n(\mathbf{X}_{1:n}) = p(\mathbf{X}_{1:n}|\mathbf{Y}_{1:n})$, or variants thereof. This causal strategy can be performed in an online framework. However, it only considers the observations up to the current frame. Therefore, there is a risk of discarding the optimal label sequence if its partial conditional probabilities $\bar{p}_n$ is low for a frame n.

We propose another strategy, which takes into account the whole signal for the pruning of the search space. It is a hierarchical approach, inspired by the FastDTW algorithm [18]. The algorithm presented is a variation of the one proposed in [41], in order to take into account the possible structure changes between the performance and the score. The idea is to first search for an alignment at a coarse level and then use the result to prune the search space at a more precise level. For this, we take advantage of musical structural units which are given by the score, namely *beat* and *measures* (or *bars*) whose variables are denoted by B and M respectively.

Since the alignments at these higher levels aim at speeding up the global process, low complexity models are preferred. Thus, we exploit Markovian CRFs, whose hidden variables

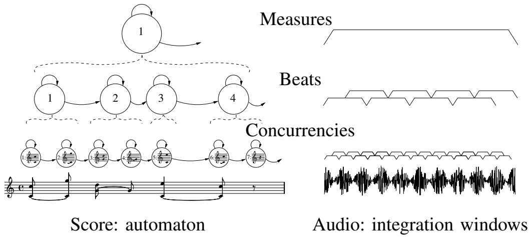


Figure 7. Label automata and integration windows (over which are calculated the observations) at the three considered levels of hierarchy. The sub-labels used in the *concurrency* level are not represented.

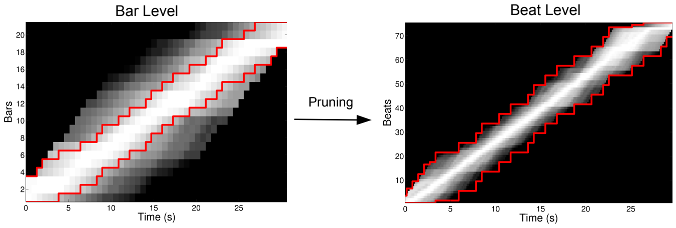


Figure 8. Principle of the hierarchical pruning method (first step). The grey scale of a cell corresponds to the value $\hat{p}_{\tilde{n}}(m)$ of (16). At the beat level, only the domain delimited by the lines is explored.

are the position labels of the corresponding level (*measure* or *beat*). Furthermore, as the corresponding labels account for larger temporal scales, the observations are extracted over longer time windows and their time resolutions (or sampling rates) are smaller than the ones used at the *concurrency* level. Any of the models presented in Section III can be used in the lowest level. Fig. 7 illustrates the label automata and the feature extraction, at the three levels of hierarchy.

The algorithm begins at the *measure* level. Let $\tilde{\mathbf{Y}}$ be the sequence of integrated observations considered at this level, where $\tilde{N}$ is the number of integration windows. A Markovian CRF as presented in Section III-A1 is used to model this observation sequence. Let $\tilde{\psi}$ and $\tilde{\phi}$ be the corresponding transition and observation functions. According to this model $\mathcal{M}$ we calculate, for every measure m and every window $\tilde{n}$, the "maximum posterior path probability" at the bar level

$$\hat{p}_{\tilde{n}}(m) = \max_{\substack{\mathbf{M}_{1:\tilde{N}} \\ M_{\tilde{n}}=m}} \left\{ p(\mathbf{M}_{1:\tilde{N}} | \tilde{\mathbf{Y}}; \mathcal{M}) \right\} \tag{16}$$

where $\mathbf{M}_{1:\tilde{N}} = M_1, \ldots, M_{\tilde{N}}$. This value can be computed by an extension of the Viterbi algorithm which can be seen as a transformation of the standard forward-backward algorithm, where the operation `sum` is replaced by `max`.

If we assume that the simplified model used at this level is consistent with the complete low-level model, and thus that the values of $\hat{p}_{\tilde{n}}(m)$ are close to the "real" probabilities, they can be used to discard low-probability label hypotheses. For each window $\tilde{n}$, we sort the measures m according to the values of $\hat{p}_{\tilde{n}}(m)$. Then, only a (small) fixed number K_{M} of measures are kept for the alignment at the lower level. The value of K_{M} is set in an adaptive way, depending on the "posterior path probabilities": it is the minimum number such that all the paths whose posterior probabilities are above a threshold are kept. This threshold is chosen as $\frac{1}{\eta} \max_{\mathbf{M}} p(\mathbf{M}|\tilde{\mathbf{Y}})$. The parameter η controls the tradeoff between accuracy and complexity. Fig. 8 illustrates this pruning process.

The same procedure is then performed at the lower level, where only the undiscarded labels are explored. At this *beat* level, another Markovian CRF is used to model the observation sequence, in order to maintain a low complexity. Further label sequences are pruned out, and the final alignment is searched for, only among the remaining label sequence hypotheses.

The difference between this pruning method and the version proposed in [41] is that in the latter, we kept the states which were inside a "tolerance radius" around the Viterbi path. However, this notion of adjacent concurrencies is no longer applicable in the case where there can be structural difference between the score and the performance.

Note that it would be possible to use the marginal probabilities $p(M_{\tilde{n}} = m|\tilde{\mathbf{Y}})$ instead of the whole sequence probabilities $\hat{p}_{\tilde{n}}(m)$ for the pruning process. However, we believe that the latter are more relevant in our problem. Indeed, the marginal label probability $p(M_{\tilde{n}} = m|\tilde{\mathbf{Y}})$ is the sum of the probabilities of all the sequences verifying $M_{\tilde{n}} = m$. Hence there is a risk that some label/observation cells corresponding to many average-score label sequences be favored compared to a cell containing an isolated high-score sequence.

B. Multi-Level Observations

In the measure and beat levels, only chroma vector features are used as observations, since the onset feature and the cyclic tempogram correspond to lower level variables and do not appear in the CRF used at these levels. These higher level chroma vectors are computed as the average of the original chroma vectors over the integration windows (represented in Fig. 7). The use of averaged observations is musically justified since the harmony (and thus the chroma information) is in general homogeneous over a whole beat (or measure) duration. The integration windows are chosen in relation to the possible tempi. In the hypothesis where the tempo is stable, one could use an estimation of the average tempo for setting the window lengths and hop-size. However, this can pose a risk in the presence of large tempo changes, since the observation sample rate must be at least as large as the corresponding beat or bar frequency. For example, if the real local tempo were faster than the beat observation sample rate, some beat labels would not be reached by the algorithm, which would result in discarding the corresponding path at lower levels.

In our case, we do not want to make any limiting assumption. Hence, the integration parameters are chosen so as to take into account the fastest acceptable tempo. We set the integration length for the beat level to 240 ms, corresponding to a very fast tempo of 250 beats per minute.A 1/3 overlap is used, yielding a 160-ms hop-size. For the measure level, the length and the time resolution of the integration window depend on the time signature and it corresponds to the number of beats in a bar. For example, for a 4/4 signature (four beats in each bar), the length is 960 ms and the hop-size is 480 ms.

The observation function used at these levels is the same as defined in Section IV-A. However, it compares averaged chroma vectors with an "integrated distribution" corresponding to each label. This distribution is the superposition of the chroma templates associated to the concurrencies that the label (e.g. a measure) contains, weighted by their durations in beat.

C. Fast Alignment at a High Level: Beat Level CRF (BLCRF)

The same strategy can be exploited in order to obtain a fast, coarse alignment, by stopping the hierarchical pruning process higher than the *concurrency* level. Hence, we introduce the Beat Level CRF (BLCRF), which only decodes the CRFs down to the beat level. The concurrency-level alignment is then deduced thanks to an interpolation of the concurrency onset times between the detected beat frontiers, assuming that the tempo is constant over each beat.

VI. Experimental Study

A. Database and Settings

1) Database: The database used in this work is composed of two corpuses. The first one contains 59 classical piano pieces (about 4h15 of audio data), from the MAPS database [42]. The recordings are renditions of MIDI files played by a Yamaha Disklavier piano. The alignment ground-truth is given by these MIDI files. The target scores are built from the same MIDI files as the ground truth. However, the tempo is fixed to a constant value such that the piece duration corresponding to this tempo is the same as the audio file length.

The second corpus is composed of 90 pop songs (about 6h) from the RWC database [43]. Aligned MIDI files are provided with the songs. These files have been aligned thanks to an automatic beat tracker and then manually corrected. In this database, the tempi are almost always constant. Thus, we introduced random tempo changes in the midi scores in order to simulate tempo variations in the performance. Each file is separated into segments of equal length in beats (about 16 beats) and for each segment, a unique tempo is randomly sampled from a uniform distribution between 40 and 240 beats per minute. These modifications represent an extreme case of tempo changes. Since pop song scores which can be found on the Internet (as in our application scenario) often contain errors in the percussion part, or may not even include a percussion part, we chose to discard the percussion in the scores.

A learning database has been created, containing one hour from each corpus, in order to evaluate the model parameters. The evaluation is then run on the remaining of both MAPS and RWC datasets.

2) Evaluation Measure: The chosen evaluation measure is the onset recognition rate, defined as the fraction of onsets which are correctly detected (*i.e.* onsets which are detected less than a tolerance threshold θ away from the ground truth onset time). The value $\theta = 300\,\mathrm{ms}$ is based on the MIREX contest. For a more precise alignment evaluation, we use two other thresholds: 100 ms and 50 ms.

3) Tested Systems: We evaluate systems using the three dependency structures exposed in Section III. The graphical representations of these models are displayed in Fig. 5. For each of these structures, the use of the neighborhood in the observation function (see Section IV-A) is assessed by comparing two versions of the system, using different values of the neighborhood parameter δ. In the first version, we have $\delta = 0$, which means that no neighborhood is taken into account. The second version uses a 1-s neighborhood.

For all systems which use a tempo variable, the set of possible tempo values is, in beats per minute:

$$\begin{aligned}\mathcal{T} =&\{28, 30, 34, 40, 48, 56, 64, 72, 80, 88, 96, 104,\\ &112, 120, 132, 146, 160, 176, 192, 208, 224, 240\}.\end{aligned} \tag{17}$$

The values of the observation function parameters are estimated on the learning dataset, thanks to a coarse grid search. The chroma parameter α is set to 10. The value of the onset parameter is $\mu_{\mathrm{f}} = 100$ for the MCRF and $\mu_{\mathrm{f}} = 10$ for the SMCRF and HTCRF models. The tempogram parameter is set to $\mu_{\mathrm{t}} = 100$ for the MCRF and to $\mu_{\mathrm{t}} = 10$ for the other models. The onset and tempo parameters have higher values in the MCRF models than in the others. This can be explained by the fact that they compensate for the lack of temporal model in the former system.

a) Markovian CRF: For the MCRF systems, we observed that the values of λ_0 and λ_1 in the transition function of (7) do not really influence the alignment results. They are set as in Section III-A1, so as to maximize the probability of the duration indicated in the MIDI score file.

In these models, the duration and tempo labels of different frames (D_{n_1}, T_{n_1}) and (D_{n_2}, T_{n_2}) are considered as independent given the other variables. Therefore, the maximizations over the processes $\mathbf{D}$ and $\mathbf{T}$ can be done "at the observation level". We define:

$$\bar{\phi}\left(c_n, a_n, \mathbf{y}\right) = \max_{d_n, t_n} \left\{ \phi_{\mathrm{c}}\left(x_n, \mathbf{v}\right) \phi_{\mathrm{a}}\left(a_n, \mathbf{f}\right) \phi_{\mathrm{t}}\left(t_n, \mathbf{g}\right) \right\}. \tag{18}$$

The label sequence $(\hat{\mathbf{c}}, \hat{\mathbf{a}})$ corresponding to the optimum of (1) is then:

$$(\hat{\mathbf{c}}, \hat{\mathbf{a}}) = \underset{(\mathbf{c},\mathbf{a})}{\operatorname{argmax}} \left\{ \bar{\phi}\left(c_1, a_1, \mathbf{y}\right) \prod_{n=2}^{N} \psi\left(x_n, x_{n-1}\right) \bar{\phi}\left(c_n, a_n, \mathbf{y}\right) \right\}. \tag{19}$$

The number of "cells" explored by the Viterbi algorithm (without any pruning process) is $2Q_{\mathrm{C}}N$, where Q_{C} is the number of possible concurrency labels (2 is the number of possible phase labels). When using our pruning algorithm, this space complexity (at the concurrency level) becomes $2\tilde{Q}_{\mathrm{C}}N$ with $\tilde{Q}_{\mathrm{C}}$ the average number of maintained concurrency labels after the pruning process.

b) Semi-Markovian CRF: For the SMCRF systems, the transition function seen in Section III-B is set so that the duration model of each concurrency is Gaussian, whose mean is the duration indicated in the MIDI score file and whose standard deviation is 65 ms.

In the decoding of these models, the maximization factorization can only be performed on the tempo variables, since the occupancy variables are no longer conditionally independent. Therefore $Q_{\mathrm{C}}Q_{\mathrm{D}}N$ cells have to be explored by the Viterbi algorithm ($\tilde{Q}_{\mathrm{C}}Q_{\mathrm{D}}N$ with pruning), where Q_{D} is the mean number of possible sub-concurrency labels.

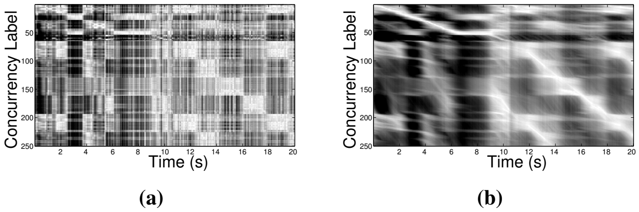
Model	MCRF		SMCRF		HTCRF	
δ	0	1 s	0	1 s	0	1 s
300 ms	93.6	94.8	97.8	98.0	99.4	99.4
100 ms	81.8	85.7	90.6	93.8	98.0	98.0
50 ms	65.8	69.5	76.2	83.7	91.3	91.8

MAPS database

Model	MCRF		SMCRF		HTCRF	
δ	0	1 s	0	1 s	0	1 s
300 ms	86.1	88.3	94.2	94.0	99.2	99.2
100 ms	59.9	65.5	75.5	79.4	94.6	94.6
50 ms	38.7	43.2	52.3	57.8	75.5	75.2

RWC-pop database

Table I
Recognition Rates (in %) obtained by the systems for different values of the threshold θ and the neighborhood parameter δ.



(a) **(b)**

Figure 9. Contributions of the instantaneous chroma observation **(a)** and of the neighborhood **(b)** in the value of the MCRF chroma observation function $\bar{\phi}_{\mathrm{C}}$ (see Section VI-A3). White indicates higher value.

c) Hidden Tempo CRF: In the case of the HTCRF, the full-fledged Viterbi algorithm explores $Q_{\mathrm{C}}Q_{\mathrm{D}}Q_{\mathrm{T}}N$ ($\tilde{Q}_{\mathrm{C}}Q_{\mathrm{D}}Q_{\mathrm{T}}N$ with pruning) cells, where Q_{T} is the number of possible tempo labels. The parameters of the transition function are set thanks to a grid search run on the learning database. Their values are chosen as: $\gamma_{\mathrm{d}} = 100$ and $\gamma_{\mathrm{t}} = 1\,000$.

d) Comment on the Duration Penalties: One may wonder why our SMCRF employs a duration model with a fixed standard deviation whereas in (9), for the case of the HTCRF, it is proportional to the expected concurrency duration. Through preliminary experiments on the training database, we observed that the "proportional strategy" led to higher recognition rates with the HTCRF model which does not exploit neighboring frames (96.6% against 94.6% with a fixed penalty for $\theta = 100\,\mathrm{ms}$). This tends to confirm our intuition that the duration deviations are proportional to the note lengths.

On the other hand, the results were different with the SMCRF: the accuracy improved on the MAPS files (from 90.8% to 94.3%), where the score durations do not differ much from the real audio ones, but it greatly decreased on the RWC songs (from 71.2% to 64.7%) where large tempo changes occurred. Indeed, the duration model becomes very rigid when the score durations are short. The duration deviation penalties would be comparatively higher for slow tempi than for fast ones. Hence, in the presence of important tempo deviations, a constant standard deviation proves more efficient.

B. Alignment Results

The recognition rates obtained by the tested systems are presented in Table I. Since the annotation of the RWC database is not perfect, the recognition rates for a 50-ms threshold are not to be fully trusted. Therefore, they are only indicative but it is worth noting that they exhibit the same tendencies. The radii of the 99% confidence intervals are smaller than 0.4%.

All the tested systems obtain higher scores on MAPS than on the RWC database, which can be explained by three main facts. First, contrary to MAPS, the RWC database contains percussive instruments (mainly drums) and other unpitched sounds (talking voices, applause…). The presence of these sounds can affect the chromagram representation as well as the onset feature. Second, RWC pieces often contain many instruments, whose relative mixing levels can be heterogeneous. Hence, some "background" instrumental parts are barely audible, and their note changes are difficult to detect. Finally, the leading voice, which is dominant in most pop songs, may contain various effects such as vibrato, pitch bend, etc… which are not described in the score.

For both neighborhood parameter settings, the accuracy increases with more intricate duration models. Indeed, the Semi-Markovian CRF system always obtains higher recognition rates than the Markovian CRF, and is outperformed by the Hidden Tempo CRF system. For example, the MCRF reaches a highest recognition rate of 94.8% for a 300-ms tolerance threshold on the MAPS database. Adding an absolute duration model is efficient, since the SMCRF obtains a 98.0% performance. Finally letting the duration model depend on a tempo variable allows the HTCRF to further increase the accuracy, and the best recognition rate is then 99.4%.

Another important observation is that taking into account the neighborhood does increase the fine-level precision of the first two systems (MCRF and SMCRF). For the MCRF system, discarding the neighborhood information comes to ignoring both the occupancy variable D and the tempo variable T. Hence, the neighborhood information corresponds to an implicit model of duration and tempo and it allows for an absolute 1% to 2% increase of the 300-ms recognition rates. The benefit of considering the neighborhood is even greater at finer levels of precision: improvements of at least 4% for tolerance thresholds of 100 ms and 50 ms are obtained. Fig. 9 compares the contributions of the instantaneous observation and of the neighboring frames on the observation function for a pop song excerpt. In this example, the neighborhood contribution visibly emphasizes the alignment path and some repetitions of the same concurrency sequence.

For the SMCRF, setting the neighborhood parameter δ to 0 means discarding the tempo variable. The addition of a dependency between the concurrency label and the tempo increases almost all the recognition rates, especially at fine levels of precision. For example on the MAPS database, the absolute improvements for 100-ms and 50-ms thresholds are respectively 3.2% and 7.5%.

There is an exception for the 300-ms recognition rate on the RWC database, which is worse (although not significantly) with the neighborhood exploitation (94.0% *vs* 94.2%). The main reason for this accuracy loss is the fact that, on a few pieces, the system does not follow the ground truth path, but a repetition of it, *i.e.* very similar concurrency sequence.

An example is displayed in Fig. 10, where the final section is an *ad libitum* repeat of the same sequence. In this example,

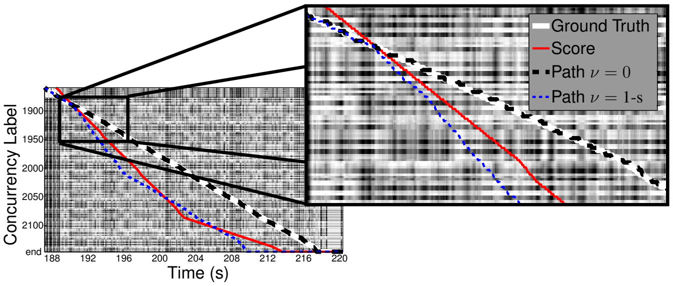


Figure 10. Example of the "wrong repetition" phenomenon: in this case, where the chroma observations are very noisy, the exploitation of the neighborhood smooths out the observation functions and the system is more likely to follow a wrong temporal model.

Pruning	Search Space		Run time	Errors
	Beats	Concurrencies	in s (% RT)	(nb)
None	–	100%	3 489 (12%)	0
BS $\nu = 600$	–	27.24%	1 330 (4.4%)	1
$\eta = 10\,000$	0.37%	15.86%	1 078 (3.6%)	0
$\eta = 1\,000$	0.32%	13.38%	813 (2.7%)	0
$\eta = 100$	0.26%	10.62%	643 (2.1%)	0
$\eta = 50$	0.24%	9.74%	617 (2.0%)	0
$\eta = 20$	0.21%	8.52%	567 (1.9%)	1

Table II
PERFORMANCE OF THE MCRF SYSTEM WITH OUR PRUNING METHOD. SEARCH SPACE IS THE RATIO OF THE LABELS EXPLORED AT EACH LEVEL, OVER THE TOTAL LABEL NUMBER (IT IS 0.15% AT THE BAR LEVEL, FOR ALL SYSTEMS). RT STANDS FOR REAL-TIME, BS IS "BEAM SEARCH".

the percussion is strong, which makes the chroma observation function very noisy. In this case, the use of the neighboring values does not help, but rather further increases the noise level. The system exploiting the neighborhood is then more easily driven by the duration model, which indicates here a faster tempo than the real one. Note that on this example, the final backtracking of the algorithm does not disambiguate the different repetitions, because the "end" concurrency label, which models silence or noise has a high value for the observation function, due to the strong percussion level. And indeed, most of the "wrong repetition" problems occur at the very end of the pop songs.

However, we believe that the importance of this phenomenon is limited, since in fact most of the detected concurrencies are equivalent to the ground truth concurrencies, and since this happens with the combination of both inaccurate chroma models and untruthful score durations.

Thanks to the explicit tempo model, the HTCRF obtains even higher results than the other systems. The "wrong repetition" problem does not occur since the duration priors are much more reliable in this case. However, the exploitation of the neighborhood does not (in most cases) improve the alignment performances. This may be explained by the fact that the tempo process is explicitly modeled. Thus, the implicit consideration of the tempo which is exploited through the neighborhood does not add much information. However, the exploitation of this observation function may lead to a small improvement of the fine-level alignment precision, since there is a 0.5% increase of the recognition rate on the MAPS database for a 50-ms threshold.

C. Pruning Performances

We tested the performance of the proposed pruning strategy for the simplest system, *i.e.* the MCRF model without use the neighborhood information ($\delta = 0$). The results on the whole test database are presented in Table II, for different values of the pruning parameter η (defined in Section V). Several values are presented: the search space is the mean number of labels which are considered in the decoding process, over the total number of labels. Run times are given as times for processing the whole database (119 pieces), excluding the feature extraction phase. The implementation of the algorithms is in MATLAB, and was run on an Intel Core2, 2.66 GHz with 3 Go RAM under Linux. The number of "pruning errors" is also presented. A pruning error occurs when the ground truth alignment path is discarded by the pruning process. We compare those results to a system without any pruning, as well as a system exploiting the *beam search* strategy, where at each step the ν "most promising" labels are maintained.

The results show the benefit of our pruning method, since the search space and run time of all the tested systems are lower than the reference systems. No pruning error occurs until a value of $\eta = 50$, whose corresponding run-time is less than 1/5 of the reference system (617 s against 3 489 s). In terms of alignment precision, down to the value $\eta = 20$, the obtained alignments are the same as for the reference system with no pruning. Hence, the reduction of complexity does not affect the alignment precision. In comparison, the beam search strategy is underperforming: indeed its pruning performance is significantly lower for a threshold leading to a single alignment error. In this particular case, the ground truth alignment path is discarded because a low partial Viterbi score is obtained on the beginning of this path. With our method, the whole signal is considered (although at a coarse level) and consequently, the risk of discarding the searched path is attenuated.

This problem of beam search would probably be less likely with the other dependency structures (SMCRF and HTCRF), because the explicit duration model would prevent the system from remaining "stuck" in the same concurrency. However, our method has another advantage compared to beam search. Indeed the latter strategy requires, at each step, a sorting of all the considered partial Viterbi scores. The cost of this process is not to be disregarded when the value of the parameter ν is high. In our method, the sorting is performed at the higher level and therefore at a significantly lower cost. At any rate, both pruning strategies are compatible and it is possible to perform beam search in a search space which has already been reduced by our hierarchical method.

D. Scalability Considerations

As seen in the last section, the addition of dependencies in the graphical model allowed for more precise alignments. However, as shown in Table III, such performance improvements are obtained with a clear increase of complexity. For a given application, the appropriate system can be chosen from this table according to the desired tradeoff between

System	Imprecision: percentile		Complexity (comparisons)	Run Time
	90th	95th		
BLCRF ($\delta = 0$)	808 ms	1 545 ms	$\tilde{Q}_{\mathrm{B}} U_{\mathrm{B}} N$	0.23% RT
MCRF ($\delta = 0$)	303 ms	525 ms	$2\tilde{Q}_{\mathrm{C}} U N$	1.7% RT
MCRF ($\delta = 1$)	255 ms	475 ms	$\tilde{Q}_{\mathrm{C}}(Q_{\mathrm{D}} Q_{\mathrm{T}} + U) N$	98% RT
SMCRF ($\delta = 0$)	152 ms	255 ms	$\tilde{Q}_{\mathrm{C}} Q_{\mathrm{D}} U N$	9.4% RT
SMCRF ($\delta = 1$)	131 ms	250 ms	$\tilde{Q}_{\mathrm{C}} Q_{\mathrm{D}}(Q_{\mathrm{T}} + U) N$	110% RT
HTCRF ($\delta = 0$)	62 ms	85 ms	$\tilde{Q}_{\mathrm{C}} Q_{\mathrm{D}} Q_{\mathrm{T}}^2 U N$	305% RT
HTCRF ($\delta = 1$)	62 ms	85 ms	$\tilde{Q}_{\mathrm{C}} Q_{\mathrm{D}} Q_{\mathrm{T}}^2 U N$	394% RT

Table III

Performance/complexity characteristics of the considered systems. Imprecision is the time-difference between the detected onsets and the ground truth. U and U_{B} are the mean numbers of transitions to each concurrency and beat label respectively. $\tilde{Q}_{\mathrm{B}}$ is the number of beat labels (after pruning).

Jumps	Forbidden		Allowed	
Score	Exact	Modified	Exact	Modified
MCRF $\delta = 0$	88.9	75.6	87.4	87.3
MCRF $\delta = 1$	90.7	77.2	89.9	89.5
SMCRF $\delta = 0$	95.6	79.1	94.4	94.2
SMCRF $\delta = 1$	95.5	79.5	94.1	93.7
HTCRF $\delta = 0$	99.3	83.7	98.5	98.2
HTCRF $\delta = 1$	99.3	83.6	98.4	98.0

Table IV

Robustness experiment: recognition rates obtained on the whole test database (RWC+MAPS) with $\theta = 300$ ms.

performance and complexity. We also tested the Beat Level CRF system presented in Section V-C. This system is not very accurate (the 300-ms recognition rates are respectively 81.90% and 63.88% on the MAPS and RWC databases), however it is very fast: its run-time is only 0.23% of the audio duration.

Since our exploitation of the neighborhood information involves calculating different observation functions for each value of (C_n, A_n, D_n, T_n), the decoding complexity of the corresponding systems are higher than the ones which do not exploit the neighborhood. Hence, the MCRF model using neighborhood information is not interesting, since the alignment is costlier than with the "instantaneous" SMCRF system for lower performances. The use of the neighborhood information in the HTCRF system also seems questionable, since there is no visible performance improvements compared to the "instantaneous" HTCRF system.

E. Robustness to Structure Change

We now examine the case where structural differences occur between the score and the audio recording. To address this situation, we modify the scores so as to introduce jumps and repeats. A repetition is created in each score by duplicating an arbitrary sequence of eight bars. A jump is also added by discarding the second instance of the longest repeated section of at least four bars, when there is one.

In the models presented in Section III, the concurrency succession is necessarily the same as in the score. Thus, the transition functions have to be modified in order to take into account possible structural differences.

We assume that we know a set of possible score positions at which "jumps" can occur in the score. This set can be indicated in the score, as it corresponds to repeat signs or other repetition symbols. It can also be the result of a structural analysis of the score, using the frontiers of the detected sections. Hence, we call these positions "segment frontiers". In this work, the segment frontiers considered in the alignment process are the same as those used for the modification of the score.

Let $\mathcal{J}$ be the set of possible jumps, that is all the transitions from a segment frontier to another one. These new possibilities are added to the transition functions, with a penalization factor $\frac{1}{2}$ compared to the transition to the following concurrency. For example, the SMCRF transition function of (8) becomes:

$$\psi\left(Y_{n+1}, Y_n\right) = \begin{cases} 1 & \text{if } (C_{n+1}, D_{n+1}) = (C_n, D_n{+}1) \\ p(L_{C_n} = Dn) & \text{if } (C_{n+1}, D_{n+1}) = (C_n{+}1, 1) \\ \frac{1}{2} p(L_{C_n} = Dn) & \text{if } (C_n, C_{n+1}) \in \mathcal{J} \text{ and } D_{n+1} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Experiments were run on both exact and modified scores, with the original systems (forbidding jumps in the score) and the new ones (allowing them). The results of these experiments are displayed in Table IV. Note that these figures are given on the whole test database (MAPS + RWC).

First, one can notice a decrease of the recognition rates (of about 1%) on the perfect scores when jumps are allowed. As in the "wrong repetition problem" exposed in Section VI-B, this is due to a few pieces where the observations poorly match the chroma templates. On these pieces, the favored path jumps to the *end* label, which accepts more or less any observation, and stays there until the end of the recording. Note that only 9 pieces over 119 are concerned by this phenomenon with the HTCRF model, all from the RWC database. A solution to this problem could be to use more robust observations.

Then, consistently with one's intuition, the precision of the alignment with all the systems decreases when the score is modified. However, whereas the performance of the original systems dramatically collapses, this reduction is relatively small, about 1% to 2% for the systems allowing jumps in the score. Furthermore, even with imperfect scores, the HTCRF obtains better recognition rates than the SMCRF with perfect scores. This shows that the chosen framework is quite robust to these score modifications.

VII. Conclusion

In this paper, we propose the use of the CRF framework to address the audio-to-score alignment problem. We show that this framework encompasses the different statistical models which have been proposed in the literature. Furthermore, it allows for the use of more flexible observation functions than the different variants of the HMM framework. In particular we introduce the exploitation of observations extracted from a whole neighborhood of each time frame and we show that this can improve the fine level precision of the obtained alignment.

We use several acoustic features characterizing different aspects of the musical content, namely harmony, note attacks and tempo. A scalable framework is proposed, involving several CRF with increasingly intricate dependency structures, for an ascending accuracy in the concurrency duration modeling. We test these models on a large-scale database of polyphonic, classical and popular music. We show that an improvement of the alignment precision is obtained with more sophisticated dependency structures. Our most elaborate system, the HTCRF, obtains a very high accuracy, since more than 95% of the onsets are detected with a finer precision than 100 ms.

We additionally show how the proposed framework can be modified in order to take into account possible structural differences between the score and the performance. Our experiments show that the accuracy loss in this case is quite small (1% to 2%) since it is lower than the difference between two systems with different dependency structures.

Furthermore, we propose a novel pruning approach, which exploits the hierarchical structure of the score for a decoding of the model in a coarse-to-fine fashion, in order to reduce the decoding complexity. We show that this strategy allows for a significant reduction of both the explored search space and the run time. Experiments also show that, with our Markovian model, our approach outperforms the beam search method. The same coarse-to-fine strategy is also exploited in order to obtain a very low-cost alignment.

Some results indicate a greater sensitivity of the observation function exploiting neighboring frames to noisy chroma observations coupled with inaccurate duration priors. This "noisiness", only encountered in the pop music database, is mainly due to the strong level of unpitched sounds (such as percussion). The exploitation of a drum separation algorithm or of an explicit noise model would constitute an interesting perspective in order to "clean" the observation function values.

The CRF framework also allows for the design of other forms of potential functions. In particular the transition functions, which are here constant, can vary as functions of the observation sequence. This would permit the use of features expressing structural relations between different frames, such as the similarity of the observations, in addition to features expressing the match between the observations and the labels. One could for example imagine a transition function which would favor self-transitions (to the same concurrency) when no variation occurs between several adjacent frames. Such features would probably be robust to some deviations between the observations and the concurrency templates, such as tuning differences or pitch imprecisions. In the case of the MCRF system, one could also imagine a transition function depending on the position of the last preceding peak in the onset detection function, for an implicit temporal modeling without the cost of decoding the occupancy variable sequence.

The parameters, which have here been set thanks to a coarse grid search, could also be learned, for example through a *maximum likelihood* estimation. However, this estimation process is very complex, and one needs a sufficiently large learning database. Finally, it is worth mentioning that the CRF framework could also be applied in a real-time context with few modifications (for example, only the past frames could be considered in the calculation of the potential function).

References

[1] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proc. ICMC*, 1984, pp. 193–198.

[2] J. D. Vantomme, "Score following by temporal pattern," *Computer Music Journal*, vol. 19, no. 3, pp. 50–59, 1995.

[3] N. Orio, S. Lemouton, and D. Schwarz, "Score following: State of the art and new developments," in *Proc. NIME*, 2003.

[4] D. Damm, C. Fremerey, F. Kurth, M. Müller, and M. Clausen, "Multimodal presentation and browsing of music," in *Proc. ICMI*, 2008.

[5] C. Raphael, "A classifier-based approach to score-guided source separation of musical audio," *Computer Music Journal*, vol. 32, no. 1, pp. 51–59, 2008.

[6] P. Cano, A. Loscos, and J. Bonada, "Score-performance matching using hmms," in *Proc. ICMC*, 1999.

[7] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 360–370, 1999.

[8] L. Grubb and R. B. Dannenberg, "Enhanced vocal performance tracking using multiple information sources," in *Proc. ICMC*, 1998.

[9] A. Cont, "Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms," in *Proc. IEEE ICASSP*, 2006.

[10] M. Puckette, "Score following using the sung voice," in *Proc. ICMC*, 1995, pp. 175–178.

[11] A. Cont, "A coupled Duration-Focused architecture for Real-Time Music-to-Score alignment," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 6, pp. 974–987, Jun. 2010.

[12] N. Montecchio and N. Orio, "A discrete filterbank approach to audio to score matching for score following," in *Proc. ISMIR*, 2009.

[13] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Proc. IEEE WASPAA*, 2003.

[14] M. Müller and D. Appelt, "Path-constrained partial music synchronization," in *Proc. IEEE ICASSP*, 2008.

[15] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proc. IEEE ICASSP*, 2009.

[16] S. Dixon and G. Widmer, "Match: a music alignment tool chest," in *Proc. ISMIR*, 2005.

[17] F. Soulez, X. Rodet, and D. Schwarz, "Improving polyphonic and poly-instrumental music to score alignment," in *Proc. ISMIR*, 2003, pp. 143–148.

[18] S. Salvador and P. Chan, "Fastdtw: Toward accurate dynamic time warping in linear time and space," in *KDD Workshop on Mining Temporal and Sequential Data*, 2004, pp. 70–80.

[19] M. Müller, H. Mates, and F. Kurth, "An efficient multiscale approach to audio synchronization," in *Proc. ISMIR*, 2006.

[20] B. Niedermayer and G. Widmer, "A multi-pass algorithm for accurate audio-to-score alignment," in *Proc. ISMIR*, 2010.

[21] C. Fremerey, M. Müller, and M. Clausen, "Handling repeats and jumps in score-performance synchronization," in *Proc. ISMIR*, 2010.

[22] A. Arzt and G. Widmer, "Towards effective 'any-time' music tracking," in *Proc. of the Starting AI Researchers' Symposium (STAIRS)*, Lisbon, Portugal, 2010.

[23] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[24] B. Pardo and W. Birmingham, "Modeling form for on-line following of musical performances," in *Proc. of National Conference on Artificial Intelligence*, 2005.

[25] L. Grubb and R. Dannenberg, "A stochastic method of tracking a vocal performer," in *Proc. ICMC*, 1997.

[26] P. Peeling, A. T. Cemgil, and S. Godsill, "A probabilistic framework for matching music representations," in *Proc. ISMIR*, Vienna, Austria, 2007, pp. 267–272.

[27] C. Raphael, "Aligning music audio with symbolic scores using a hybrid graphical model," *Machine Learning Journal*, vol. 65, pp. 389–409, 2006.

[28] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Computer Science Division, UC Berkeley, Jul. 2002.

[29] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." in *Proc. ICML*, 2001.

[30] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram – a mid-level tempo representation for music signals," in *Proc. IEEE ICASSP*, Mar. 2010.

[31] W. P. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. A. Bartsch, D. Mazzoni, C. Meek, M. Mellody, and W. Rand, "Musart: Music retrieval via aural queries," in *Proc. ISMIR*, 2001, pp. 73–81.

[32] P. Smyth, “Belief networks, hidden markov models, and markov random fields: a unifying view,” *Pattern Recognition Letters*, vol. 18, pp. 1261–1268, 1998.

[33] H. M. Wallach, “Conditional random fields: An introduction,” Department of Computer and Information Science, University of Pennsylvania, Tech. Rep. MS-CIS-04-21, 2004.

[34] S. Fine and Y. Singer, “The hierarchical hidden markov model: Analysis and applications,” in *Machine Learning Conf.*, 1998, pp. 41–62.

[35] M. T. Johnson, “Capacity and complexity of hmm duration modeling techniques,” *IEEE Signal Processing Lett.*, vol. 12, no. 5, pp. 407–410, 2005.

[36] A. Friberg and J. Sundberg, “Time discrimination in a monotonic, isochronous sequence.” *Journal Acoust. Soc. Am.*, vol. 98, pp. 2525–2531, 1995.

[37] A. T. Cemgil, H. J. 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.

[38] Y. Zhu and M. Kankanhalli, “Precise pitch profile feature extraction from musical audio for key detection,” *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 575–584, Jun. 2006.

[39] C. Joder, S. Essid, and G. Richard, “A comparative study of tonal acoustic features for a symbolic level music-to-score alignment,” in *Proc. IEEE ICASSP*, 2010.

[40] M. Alonso, G. Richard, and B. David, “Extracting note onsets from musical recordings,” in *Proc. IEEE ICME*, 2005.

[41] C. Joder, S. Essid, and G. Richard, “An improved hierarchical approach for music-to-symbolic score alignment,” in *Proc. ISMIR*, Utrecht, Holland, Aug. 2010.

[42] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Trans. Audio, Speech, Language Processing*, vol. 18, no. 6, pp. 1643 –1654, Aug. 2010.

[43] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical, and jazz music databases,” in *Proc. ISMIR*, 2002, pp. 287–288.

Cyril Joder received the engineering degree from the École Polytechnique and Telecom ParisTech, and the M.Sc. degree in acoustics, signal processing and computer science applied to music from the university Pierre et Marie Curie, Paris, France, in 2007. He is currently pursuing a Ph.D. at the department of Signal and Image Processing at Telecom ParisTech. His research mainly focuses on the applications of machine learning to audio signal processing.

Slim Essid is an Associate Professor at the Department of Image and Signal Processing-TSI of TELECOM ParisTech with the AAO group. He received the state engineering degree from the École Nationale d’Ingénieurs de Tunis, Tunisia, in 2001, the M.Sc. (D.E.A.) degree in digital communication systems from the École Nationale Supérieure des Télécommunications, Paris, France, in 2002, and the Ph.D. degree from the Université Pierre et Marie Curie (Paris 6) after completing a thesis on automatic audio classification. He has published over 40 peer-reviewed conference and journal papers with more than 50 distinct coauthors, and has served as a program committee member or as a reviewer for various audio and multimedia conferences and journals, for instance, IEEE transactions on Audio, Speech, and Language Processing, on Circuits and Systems for Video Technology, on Multimedia and EURASIP Journal on Audio, Speech, and Music Processing. He has been involved in various French and European research projects among which are Quaero, Infomgic, NoE Kspace and NoE 3DLife.

Gaël Richard received the State Engineering degree from Telecom ParisTech, France (formerly ENST) in 1990, the Ph.D. degree from LIMSI-CNRS, University of Paris-XI, in 1994 in speech synthesis, and the *Habilitation à Diriger des Recherches* degree from the University of Paris XI in September 2001. After the Ph.D. degree, he spent two years at the CAIP Center, Rutgers University, Piscataway, NJ, in the Speech Processing Group of Prof. J. Flanagan, where he explored innovative approaches for speech production. From 1997 to 2001, he successively worked for Matra, Bois d’Arcy, France, and for Philips, Montrouge, France. In particular, he was the Project Manager of several large scale European projects in the field of audio and multimodal signal processing. In September 2001, he joined the Department of Signal and Image Processing, Telecom ParisTech, where he is now a Full Professor in audio signal processing and Head of the Audio, Acoustics, and Waves research group. He is a coauthor of over 80 papers and inventor in a number of patents. He is also one of the experts of the European commission in the field of speech and audio signal processing. Prof. Richard is a member of the EURASIP and an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.