Learning Optimal Features for Polyphonic Audio-to-Score Alignment

Cyril JODER, Slim ESSID, Gaël RICHARD

***Abstract*—This paper addresses the design of *feature functions* for the matching of a musical recording to the symbolic representation of the piece (the score). These feature functions are defined as dissimilarity measures between the audio observations and template vectors corresponding to the score. By expressing the template construction as a linear mapping from the symbolic to the audio representation, one can learn the feature functions by optimizing the linear transformation. In this paper, we explore two different learning strategies. The first one uses a best-fit criterion (minimum divergence), while the second one exploits a discriminative framework based on a Conditional Random Fields model (maximum likelihood criterion).**

We evaluate the influence of the feature functions in an audio-to-score alignment task, on a large database of popular and classical polyphonic music. The results show that with several types of models, using different temporal constraints, the learned mappings have the potential to outperform the classic heuristic mappings. Several representations of the audio observations, along with several distance functions are compared in this alignment task. Our experiments elect the symmetric Kullback-Leibler divergence. Moreover, both the spectrogram and a CQT-based representation turn out to provide very accurate alignments, detecting more than 97% of the onsets with a precision of 100 ms with our most complex system.

I. Introduction

In many automatic music analysis tasks, such as audio-to-score alignment [1], automatic transcription [2], main melody extraction [3] or chord recognition [4], one needs to match the audio information (or a low-level representation directly extracted from it) with a symbolic description of the music.

In this paper, we focus on the audio-to-score alignment problem, which consists in synchronizing an audio recording of a musical piece with symbolic score. In a real-time context, this task is known as *score following* [5], which achieves the tracking of a musician's performance. Such a tracking allows for the automation of some processes to be synchronized with the performer, for example "hands-free" page turning [6] or synthetic accompaniment [7] of a live soloist. On the other hand, off-line audio-to-score matching is expected to be more precise, and can be applied to multi-modal browsing of musical pieces [8], automatic identification of musical works [9] or even informed source separation [10], [11].

An audio-to-score alignment system relies in particular on a measure of the "instantaneous match" between each audio observation and each position in the score. In many works, this correspondence is evaluated by a template-based approach, where observations are directly compared to template vectors corresponding to the score [12]. Even in this framework, the template design has seldom been addressed and most systems resort to heuristic forms [13], [14]. İzmirli and Dannenberg [15] study a similar construction in the case where both the score and the audio observations are transformed into a "chroma-like" 12-dimension space.They show that the classic "canonical mapping" is not the most effective one for the task of discriminating aligned and non-aligned frames. However, their work is focused on one special type of representation, since the dimension is fixed to 12, and the evaluation is performed on a classification task.

In the present paper, we extend the work reported in [16] and explore the automatic learning of the templates on several common representations of the audio signal, as well as several distance functions for their comparison. The obtained low-level layers are evaluated in an audio-to-score alignment task, by integrating them into Conditional Random Fields (CRF) models as in [17]. The experiments, conducted on a large database of popular and classic polyphonic music, using two different temporal models, show that the learning of the mapping can significantly improve the accuracy of an alignment system. Furthermore, we propose two different learning strategies: a best fit criterion (minimum divergence) or a discriminative criterion, which takes advantage of the CRF model employed (maximum likelihood). We compare the efficiency of these approaches, and experimentally show that the discriminative strategy has the potential to reach a finer level of precision.

The rest of this paper is organized as follows. The global structure of an alignment system is presented in Section II, and Section III exposes the form of the template-based matching measure. Heuristic mappings used in the literature are detailed in IV, before our strategies for the learning of the templates are proposed in the following two sections. Finally, we evaluate the impact of these mappings on the alignment accuracy of two state-of-the-art systems in Section VII, and conclude in suggesting some perspectives.

II. Structure of an Audio-to-Score Alignment System

Audio-to-score alignment systems are traditionally split into two main layers. A low-level layer first calculates features

When writing this article, Cyril Joder was with the Technical University Munich, Germany. He now works at the European Patent Office in Munich (e-mail: cyril.joder@epo.org). Slim Essid and Gaël Richard are with the Institut TELECOM / TELECOM ParisTech, 37 rue Dareau, 75014 Paris, France (e-mail: {slim.essid, gael.richard}@telecom-paristech.fr)

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

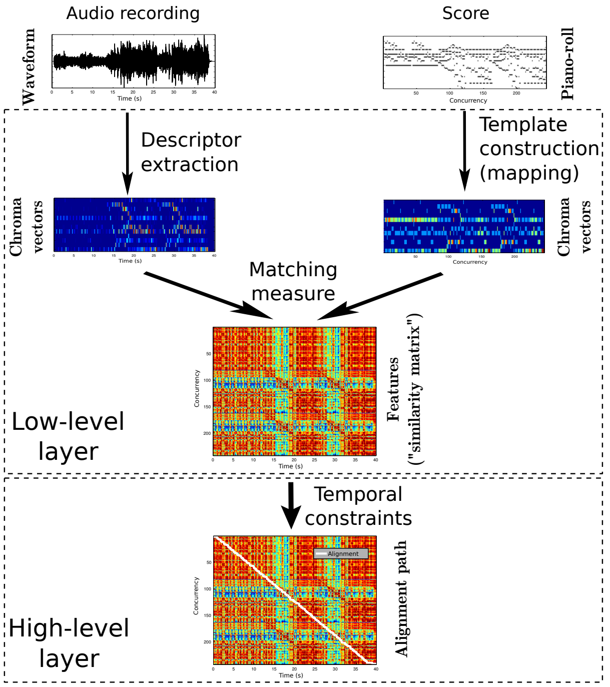


Figure 1. Structure of an audio-to-score alignment system. The audio and the score are converted into vectors of the same domain (here chroma vectors). Features are calculated by comparing these vectors, and then combined with temporal constraints by the high-level layer to perform the alignment.

associated to each element of the symbolic representation (i.e. position in the score), for each audio frame observation. Note that, following [18], the word *feature* denotes in this work a value which characterizes the correspondence between a score position and an audio observation. They distinguish from the audio *descriptors*, which characterize some properties of the audio observations alone. These local matching measures are then used by a high-level layer, which incorporates possible constraints or penalties on the temporal evolution of the score position. These constraints are designed to favor a smooth progression in the score. Finally, the output of the system is a sequence of score positions which locally match the audio observation and whose rhythmic structure conforms to the indications of the score. Figure 1 summarizes the global structure of an audio-to-score alignment system.

The alignment systems of the literature can be divided into two main groups, depending on their high-level layers. In the first one, the alignment is searched for by minimizing a cumulative cost function, based on the local matching measure, using dynamic programming techniques. This group encompasses early works on real-time score following as well as most off-line systems (for example [1], [19]). The Dynamic Time Warping (DTW) algorithm is quite extensively used [13], [20], [21], since it is efficient and computationally simple. In these systems, the feature measuring the correspondence between a symbolic element and an audio frame is calculated as a distance between the audio descriptor and a template associated to the symbolic representation.

The systems of the second group are based on probabilistic models, in particular Hidden Markov Models (HMMs), which consider the score positions as hidden random variables [22], [23]. Recently, other structures have been proposed in order to better model the note durations by introducing additional hidden variables representing the tempo [12], [24]. These models are generative: the high-level layer corresponds to the prior model, which determines the prior probability of each symbolic sequence. Hence, the low-level layer calculates the conditional probability of each observation, given each score position. However, probably because of the high number of possible note combinations in a polyphonic musical score, an estimation of these conditional probability distributions has been seldom considered. To our knowledge, only [25] and [26] describe a learning of observation distributions in the context of audio-to-score alignment, and they are limited to monophonic music. Nevertheless, most of the systems exploit heuristic forms for the conditional probabilities, which often boil down to the use of some distance between the observation and a template, as in the dynamic programming systems (for example [24]). In a previous paper [17], we showed that these models can be transposed into the Conditional Random Fields (CRF) framework, which is a class of discriminative undirected graphical models. One of the main advantages of CRFs over HMMs is the possibility to use a more flexible low-level layer. Indeed, in such a model, any feature can be employed, as it does not need to have the form of a conditional probability distribution.

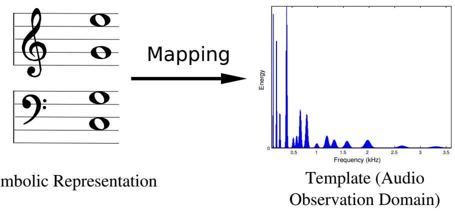


Figure 2. Creation of a concurrency template as a mapping from the symbolic to the audio observation domain (here a power spectrum representation).

The templates used in the literature are often constructed as the superposition of elementary templates corresponding to single notes. As we will see in Section III-A, this can be seen as the result of a linear mapping from what we call the “pitch vector”, containing the number of notes played at each pitch value, to the observation domain.

As far as the template creation is concerned, two main strategies can be followed. Some works use an audio synthesis of the score and extract the corresponding score observations [27]. However, it has been reported by the same authors [13] that a direct mapping from the symbolic domain to the observation domain has little impact on the alignment results using the chroma (or pitch class) representation, while avoiding the computational cost of the MIDI synthesis. Figure 2 illustrates this approach, which directly associates to each concurrency a template vector in the same domain as the audio descriptors. This is the strategy that we are interested in.

III. The Feature Function

A. General Form

For the matching of an audio recording with a symbolic representation, the low-level layer is intended to quantify the "instantaneous match" between each frame of the recording and each element of the symbolic description. We focus here on a comparison performed on the basis of the instantaneous pitch content, i.e. the notes which are currently played. As in [17], we adopt a linear representation of the score as a sequence of *concurrencies*, defined as the units of constant pitched content (sometimes also referred to as *chords). Hence, the audio observations are to be matched with the concurrencies, which are associated to each position in the score*[1].

As represented in Figure 1, the features are calculated by first creating template vectors, which correspond to the score concurrencies and are in the same domain as the audio descriptors. These templates can then be compared to the audio observations by a simple distance function. The concurrency templates are constructed as the superposition of elementary vectors, associated to each note of the concurrency. As we will see, this can be expressed as a linear mapping from what we call the "pitch vector" representation of the concurrencies.

Let us first define this representation. Assuming that the range of a musical piece does not exceed the range of the grand piano (from A0 to C8), we number the possible pitches from 1 to 88, following the chromatic scale. The pitch vector h_c of a concurrency c is defined as an 88-dimension vector whose components are the number of notes of the corresponding pitches in the concurrency. Fig. 3 illustrates the construction of this pitch vector representation. In some cases where the score also provides loudness factors for the notes (such as MIDI files), the value of the pitch vector could also be defined by these factors, as in [13]. However, we choose to use the number of notes, so as to simulate the case where the midi files result from a graphical score (as an export of a score editor, or the output of an optical music recognition system [28]). In order to take into account the portions of the signal where no note is played (in the case of silence or unpitched sounds), we introduce an additional component on the pitch vector, which is equal to 1 if and only if all the other notes are inactive. Thus, the dimension of a pitch vector is $J = 89$.

Now, let $\mathbf{W}$ be the matrix whose columns are the elementary single-note templates. As mentioned earlier, the template u_c corresponding to the concurrency c is the superposition of the elementary templates associated to the notes of c. This can be expressed in the form of a matrix multiplication:

$$u_c = \boldsymbol{W} h_c. \tag{1}$$

Hence, the matrix $\mathbf{W}$ operates a linear mapping from the pitch vector domain to the observation domain.

Let v_n be an observation vector for frame n, representing the short-time frequency content over this frame. The value of the concurrency feature $f(c, v_n)$ for concurrency c and observation v_n has the form:

$$f(c, v_n) = D\left(v_n, \mathbf{W} h_c\right), \tag{2}$$

where $D(\cdot,\cdot)$ is some distance or divergence function, $\mathbf{W}$ is a $I \times J$ matrix, I being the dimension of the observation vectors.

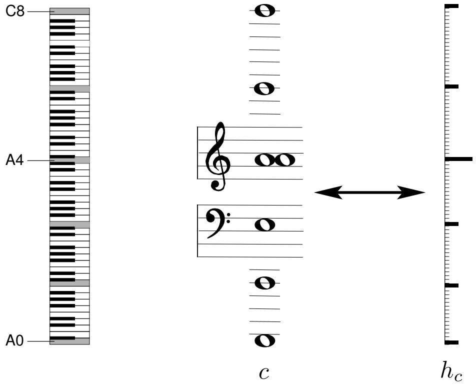


Figure 3. Illustration of the *pitch vector* representation. Left: pitch range, in the form of a grand piano keyboard; Middle: graphical representation of a concurrency, in western classical notation; Right: pitch vector representation of the same concurrency. Note that the notes of the concurrency may be played by different instruments, therefore there can be several notes of the same pitch (A4 in this example).

B. Relation with a Generative Model

This form can also be related by the generative probabilistic model exposed in [29]. Indeed, let us assume that an observation vector is the superposition of independent random vectors corresponding to the active notes. Let us suppose a Poisson distribution for each (independent) component of these one-note random vectors and let $\mathbf{W}_{i,j}$ be the distribution parameter for the component i of pitch j. The distribution parameters along the observation bins, given a note of pitch j, are then the values of the j^{th} column of $\mathbf{W}$, denoted by $\boldsymbol{W}_{:,j}$. Let h_j be the pitch vector corresponding to this single note. Since its values are: $h_j(k) = \delta_{k,j}$, where δ denotes the Kronecker delta function, the vector of parameters can be written as:

$$\boldsymbol{W}_{:,j} = \boldsymbol{W} h_j. \tag{3}$$

A sum of Poisson variables also follows a Poisson distribution, whose parameter is the sum of the individual parameters. Thus, the observations on each bin, given a concurrency c, follow independent Poisson distributions, and the parameters are the sum of the one-note parameters. Let u_c be the vector of parameters corresponding to this concurrency. The value of u_c is then given by: $u_c = \boldsymbol{W} h_c$, as in (1). Let V be a random variable representing an observation vector[2]*. The overall probability distribution of V, given the played concurrency, is then:*

$$\begin{aligned} P(V|c) &= \prod_{i=1}^{I} e^{-u_c(i)} \frac{u_c(i)^{V(i)}}{\Gamma(V(i)+1)} \\ &= \exp\left\{-D_{\mathrm{KL}}\left(V \middle\| \mathbf{W} h_c\right) + Z(V)\right\} \end{aligned} \tag{4}$$

where $Z(V)$ is a term depending only on the observation vector V, Γ denotes the gamma function and D_{KL} is the generalized Kullback-Leibler divergence. If we choose this particular divergence as the distance function D of (2), the conditional probability of (4) can be written as:

$$P(V|c) \propto e^{-f(c,V)}. \tag{5}$$

[1] In this works, the scores used are MIDI files, which explicitly specify the position of each played note. Thus, ornaments like trills or mordents are not taken into account as such, but as explicit sequences of notes.

[2] As a convention in this paper, capital letters denote random variables and the corresponding lower case letters denote realizations of the variables.

C. Distance Functions

Any dissimilarity function (which we will call distance, even if it is not a proper metric, in the mathematical sense) can be used as the matching measure of (2). In the present work, we investigate different versions of the generalized Kullback-Leibler (KL) divergence. This choice is motivated by the generative model of Section III-B. Moreover, the results of a previous study [30] as well as some preliminary tests have shown that other distances, including the Itakura-Saito divergence and the cosine distance did not outperform the KL divergence.

The first version already presented in (4), is referred to as “KL1”. Its expression is :

$$D_{\mathrm{KL}}\left(v\|u\right) = \sum_{i=1}^{I} v(i)\log\left(\frac{v(i)}{u(i)}\right) - v(i) + u(i). \qquad (6)$$

Note that, in order to make the concurrency feature robust to signal level dynamics, the observation and pitch vectors are normalized, so that the sum of the components is unity. However, we do not constrain the columns of $\mathbf{W}$ to be normalized, as this would result in a more complex (non-convex) estimation problem in Section V. This is why we use the generalized version of the KL divergence.

The formulation of (2) does not require the distance function to correspond to a generative model. Hence, we test the symmetric counterpart of the KL1 distance, referred to as “KL2”, whose expression is

$$D_{\mathrm{KL}}\left(u\|v\right) = \sum_{i=1}^{I} u(i)\log\left(\frac{u(i)}{v(i)}\right) - u(i) + v(i). \qquad (7)$$

Finally, we test the symmetric version of the divergence, denoted by “KLs”.

$$D_{\mathrm{KLs}}\left(v,u\right) = D_{\mathrm{KL}}\left(v\|u\right) + D_{\mathrm{KL}}\left(u\|v\right). \qquad (8)$$

D. Pitch Representations Used

Three types of observations have been used in the audio alignment literature in order to characterize the musical content of each frame of the audio signal, namely power spectrum, ‘semigram’ and ‘chromagram’ representations. They are summed up in Table I. In this work, the musical recordings were sampled at 16-kHz. The observations used here are computed with a 20-ms hop-size, in order to have a fine temporal resolution.

1) Power Spectrum: The Short-Term Fourier Transform (STFT) of the audio signal is used in many score following works [31], [12], [24], because of the low complexity of this transform. In this work, we exploit the power spectrum drawn from the STFT calculated on 100-ms windows. In order to reduce noise due to percussion in the high and low frequencies, we only consider the frequencies between 100 Hz and 4 kHz.

2) Semigram Representation: The semigram representation [15] is a spectrum representation with logarithmically spaced frequency bins corresponding to the semitones of the musical scale (12 bins per octave). Two methods for calculating this representation are tested here. The first one, called FilterBank SemiGram (FBSG) consists of the short-term energy at the output of elliptic filters as in [32].

Acronym	Meaning
PS	Power Spectrum
FBSG	FilterBank Semigram
CQTSG	CQT Semigram
MPCP	Müller’s PCP (from filterbank)
ZPCP	Zhu’s PCP (from CQT)

Table I
SUMMARY OF THE PITCH REPRESENTATIONS TESTED.

We also use the magnitude of a constant Q transform (CQT), with a quality factor set to one semitone. In this case, in order to maintain a good temporal precision, the values corresponding to the two lowest octaves are not computed. The longest transform is then limited to about 170-ms length, corresponding to a frequency of 100 Hz. We also limit the highest frequency bin to 4 kHz. This representation is referred to as CQT Semigram (CQTSG).

3) Chromagram Representations: Chromagram (also called Pitch Class Profile) is probably the most popular representation for offline audio-to-score and audio-to-audio synchronization [13], [1]. It consists of a 12-component vector corresponding to the spectral energies of the 12 musical pitch classes (A, A#,…). Many methods have been proposed to calculate such representations and two of them are selected in this work, based on the results of our previous study on low-level descriptors [30]. The first one, proposed by Müller [32] is the integration of the FBSG features over the different octaves. The second chroma representation is calculated according to Zhu’s method [33], which performs a peak-picking on the CQT, and then sums the amplitudes corresponding to all the octaves. These representations are denoted respectively by MPCP (for Müller’s Pitch Class Profile) and ZPCP (Zhu’s).

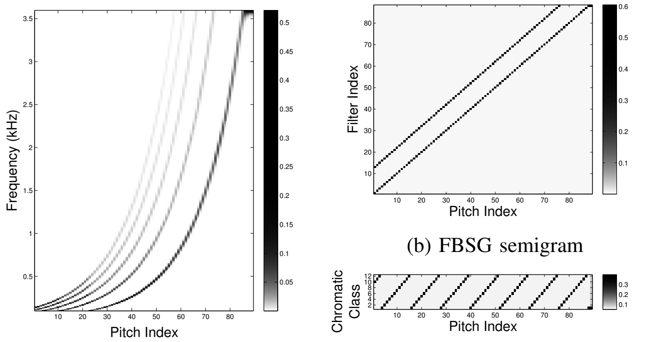
IV. Heuristic Templates

In the music-to-score alignment literature, the concurrency templates are built by following a simple heuristic. We detail here the heuristic templates that we retain in this work, for the three types of representation presented above.

A. Chromagram Templates

The heuristic chroma vector templates are derived from the canonical mapping from the pitch domain to the chroma domain [15]. For a pitch number j (as defined in Section III-D), let $\mathrm{pc}(j)$ be the pitch class of j, that is the index of the corresponding chromatic class (C, C#, …, B) in the chroma vector representation. The one-note template of pitch j is a binary template whose only non-zero component is the $\mathrm{pc}(j)$-th. The template is superposed to a uniform distribution accounting for noise. This uniform component can also be seen as a smoothing filter, preventing zeros in the templates (which would be a problem with the divergence used). The importance of this noise term is controlled by the parameter $q \in [0,1)$. The values of the matrix W are then:

$$\mathbf{W}_{i,j} = (1-q)\delta_{i,\mathrm{pc}(j)} + \frac{q}{I},. \qquad (9)$$



(b) FBSG semigram

(a) Spectrogram

(c) ZPCP chromagram

Figure 4. Heuristic mapping matrices for three of the representations. **(a)**: spectrogram; **(b)**: FBSG semigram; **(c)**: ZPCP chromagram. The values of the matrix coefficients are represented as gray levels.

B. Semigram Templates

In the case of the semigram representation, the mapping used is very straightforward: the non-zero components of the binary template for pitch j correspond to the first harmonics of the note, as in [34]. In this work, we use two harmonics. Hence the matrix W is defined as:

$$\mathbf{W}_{i,j} = \frac{(1-q)}{2}\left(\delta_{i,j}+\delta_{i,j+12}\right)+\frac{q}{I}. \tag{10}$$

C. Power Spectrum Templates

For the power spectrum representation, the templates are constructed as in [12]. A pitch is represented as a Gaussian mixture whose components correspond to the first K harmonics. Formally, let $b(j)$ be the fundamental frequency of the pitch j, expressed in the scale of the STFT bins. We write:

$$\mathbf{W}_{i,j} = (1-q)\sum_{k=1}^{K} w_k \mathcal{N}\left(i; kb(j), \sigma_{j,k}^2\right)+\frac{q}{I}, \tag{11}$$

where $\mathcal{N}(\cdot;\mu,\sigma^2)$ denotes the normal density function with mean μ and variance σ^2. The weight parameters w_k are proportional to $1/k^2$ and scaled so that $\sum_{k=1}^{K} w_k = 1$. The "bandwidth" parameters $\sigma_{j,k}^2$ are set to 30 cents (30% of a semitone) and we consider $K = 5$ harmonics.

For all these representations, the "noise template", corresponding to the absence of pitched sound, is set to the uniform value $\frac{1}{I}$.

D. Estimation of the smoothing parameter

The "learning" of these heuristic projection matrices here only consists in a setting of the smoothing parameter q of equations (9) to (11). This can be done by a grid search : alignments are computed on a training database, using a set of possible values and the value leading to the highest performance is chosen.

V. Minimum Divergence (MD) Learning

The heuristic templates presented in the previous subsection may seem somehow arbitrary. Indeed, since they are heuristic, the main motivation for the chosen values of the parameters is the fact that they give good results in an audio-to-score alignment application. This is the reason why we now address the problem of learning the matrix $\mathbf{W}$controlling the mapping from the pitch to the observation domain, using a database of real aligned music.

A. Formulation

For the training process, a grid search strategy would be intractable, due to the dimensionality of the problem. Therefore, another criterion than the alignment performance has to be chosen for determining the optimal value of $\mathbf{W}$. As already mentioned in Section III-A, the formulation of (2) with the KL1 divergence function can be derived from a generative Poisson model for each note. In this generative model, maximizing the log-likelihood of the ground-truth concurrency sequence is equivalent to minimizing the cumulative divergence between observations and templates along this sequence.

Following this idea, we adopt the Minimum Divergence (MD) criterion. For each musical piece s of the training set, let N_s be the total length of the piece, in number of frames. Let $\underline{v}_{1:N_s}^s = v_1^s \dots v_{N_s}^s$ and $\underline{c}_{1:N_s}^s = c_1^s \dots c_{N_s}^s$ be respectively the pitch observations and the ground-truth concurrencies of this sequence. For notation simplicity, we write $h_n^s = h_{c_n^s}$ for the pitch vectors corresponding to the annotated concurrencies. The optimal matrix $\hat{\mathbf{W}}^{MD}$ is then defined by:

$$\hat{\mathbf{W}}^{MD} = \operatorname*{argmin}_{\mathbf{W}} \sum_{s} \sum_{n=1}^{N_s} D\left(v_n^s, \mathbf{W}h_n^s\right). \tag{12}$$

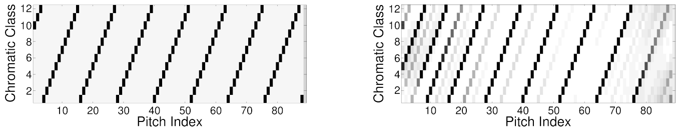
The obtained cost function is convex if the distance function used is convex. Hence, with the three divergences presented in Section III-C, we have a convex minimization problem, which can be solved by numerous strategies. The chosen iterative algorithm is a variant of Newton's method, based on the trust region *concept, in which the inversion of the Hessian is approximated by the method of [35]. We exploited the implementation of the* optimization toolbox *for MATLAB. The initialization point of the algorithm was the heuristic template, whose value of the smoothing parameter was determined by a grid search as in Section IV-D. The optimization algorithm stopped when the decrease of the objective-function was smaller than 10^{-6}, or when the absolute variation of the norm $\|\mathbf{W}\|$ was smaller than the same threshold.*

The stopping conditions correspond to variations of the objective-function or of the norm $\|\mathbf{W}\|$ being less than 10^{-6}.

B. Training and Evaluation Database

In this work, we use two datasets. The first one contains 59 classical piano pieces (about 4 h 15 of audio data), from the MAPS database [2]. The recordings are renditions of MIDI files played by a Yamaha Disklavier piano. The alignment ground-truth is given by these MIDI files. The second corpus consists of 90 pop songs (about 6 h) from the RWC database [36], with aligned MIDI scores. Since the percussion track of the MIDI files often contain errors, we choose to discard the percussion in the scores.

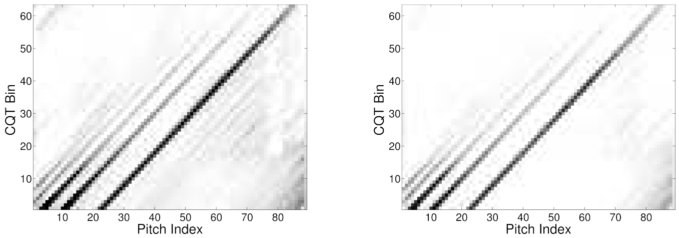
The training database is composed of 50 randomly selected pieces (220 min), 20 from MAPS and 30 from the RWC corpus.



(a) Heuristic matrix

(b) Matrix learned by MD criterion with KLs distance

Figure 5. Comparison of two mapping matrices, for the MPCP representation. The gray scale is the same on both images.



(a) KL1 distance

(b) KL2 distance

Figure 6. Comparison of two mapping matrices learned by the MD criterion with different distance functions, for the CQTSG representation. The gray scale is the same on both images.

Distance	KL1		KL2		KLs	
Mapping	H	MD	H	MD	H	MD
PS	57.7	67.9	55.5	13.6	66.3	69.9
CQTSG	63.1	67.1	63.6	67.9	64.9	68.2
FBSG	55.8	59.7	59.2	61.5	60.4	61.7
ZPCP	56.7	58.4	56.6	58.7	56.9	58.6
MPCP	51.4	53.5	51.8	53.8	52.4	54.6

Table II
RECOGNITION RATES (IN %) OBTAINED WITH HEURISTIC (H) AND MD-LEARNED MAPPINGS (MD) FOR THE TESTED DISTANCES AND REPRESENTATIONS.

In order to reduce overfitting to specific pitches or keys, 12 versions of each piece are used in the training process, by jointly transposing the observations and the pitch vectors up to -6 and $+5$ semitones. Thus, the number of training samples for a pitch template is homogeneous over a whole octave. This transposition is performed by a circular permutation for the chromagram representation, a simple 'shift' of the values in the case of the semigram representation and a frequency scaling for the spectrogram. In the latter case, the new 'scaled frequency bins' do not always correspond to original frequencies. Therefore, the values affected to these new bins are estimated by a linear interpolation of the spectrogram. The remainder of both datasets is used for the evaluation.

C. Results

1) Obtained Mapping Matrix: The learned mapping matrix for the MPCP representation with the KLs distance is displayed in Fig. 5 and compared with the corresponding heuristic mapping. In this example, it is visible that the one-note templates (i.e. the rows of the matrix) select not only the fundamental frequency and the first harmonic as the heuristic templates do, but also higher partials. Moreover, the weights given to these partials are not uniform since they depend on the note. One can also observe that for the lowest pitches, higher weights are allocated to the high partials. This can be explained by a greater energy in the higher partials of the low notes, but also by the correlation between the notes in the training database. Indeed, low notes often correspond to the bass of a chord. Hence they frequently occur concurrently with notes corresponding to their harmonic partials, resulting in heavier weightings of these partials. These behaviors are common to all the settings.

However, the three distance functions do not result in exactly the same mapping matrices. The results of the minimisation of the KL1 and KL2 distances, for the CQTSG representation are compared in Fig. 6. One can first notice that bins corresponding to lower octaves are selected for the highest notes. This is due to the phenomenon just described. Indeed, high pitches often occur concurrently with the lower octave. Thus, the presence of this lower octave is learned in the template. Another observation is that the absolute weights of the templates are almost always greater after learning with the KL1 distance than with KL2. This is due to the fact that the KL1 version (6) strongly penalizes the templates bins $u(i)$ whose values are small compared to the observation $v(i)$, as pointed out in [4]. Thus, the templates learned with KL1 tend to be overestimated. Symmetrically, KL2-learning tends to underestimate the values of the mapping matrix and the KLs version constitutes a trade-off between both behaviors.

2) Alignment Accuracy with a simple System: We evaluate the influence of the low-level layers in an alignment task. To this purpose, a simple strategy is chosen. Given the sequence of observation vector $\underline{v}_{1:N}$ (of length N) corresponding to the audio recording, the alignment is performed by searching for the optimal concurrency sequence $\hat{\underline{c}}_{1:N}$, defined by:

$$\hat{\underline{c}}_{1:N} = \operatorname*{argmin}_{\underline{C}_{1:N}\in\mathcal{C}} \sum_{n=1}^{N} f\left(C_n, v_n\right), \tag{13}$$

where $\mathcal{C}$ is the set of acceptable concurrency sequences, that is the concurrency sequences following the same order as in the score. The optimal sequence can be easily computed thanks to a dynamic programming technique. This simple method is not expected to provide very precise alignment, since it does not take into account any duration information. It is rather intended to emphasize the differences between the low-level layers tested.

The precision of an alignment is evaluated using the alignment 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). In this paper, the results are presented for $\theta = 100\,ms$. However, the relative behaviors of the different systems have been observed to be the same for values $\theta = 300\,ms$ and $\theta = 50\,ms$: only the absolute values of the alignment rate change.

The obtained alignment rates are displayed in Table II. The number of onsets in the test set is approximately 120 000. It is clear from these results that learning the mapping matrix does improve the alignment accuracy. Indeed, for all the tested settings except one, the alignment rates significantly increase compared to the heuristic templates.

The only exception is the case of the power spectrum representation with the KL2 divergence, where the alignment rate drops from 55.5% to 13.6%. This can be explained by the bias of the "no-note template" learned with this distance: as already mentioned, the KL2 divergence strongly penalizes template values which are small compared to the observations. Thus, as the power spectrum observations associated to the noise template are very diverse, the learning process tends to reduce the values of all the bins. This results in a bias toward this template, which exhibits a relatively small distance with virtually any observation. As a consequence, the obtained alignments tend to be "stuck" in the initial or final noise states. This problem does not appear with the other features, probably because of the dimension reduction, which reduce the discrepancy between the observations of the "noise state" as well as the integration of the energy over relatively large frequency bands, which generally prevents the feature values from being too small.

As already mentioned, the KLs distance seems to operate a good trade-off between the biases of both KL1 and KL2 divergences. Hence, for each representation, it always performs at least as well as the other distances. We will then consider only this distance in the rest of our experiments.

We can also compare the performances of the tested representations. The best results are obtained by the power spectrum representation (69.9%). Then the semigram representations induce a higher accuracy than the chromagrams. This is due to the reduction of the dimensionality and the fact of discarding the octave data, which entail a loss of useful information. Nevertheless, the chroma representation may still be useful in the case of scores which are not truly reliable, since it has the potential for improved robustness to octave errors, as shown in [30] where the database contains such errors. Finally, the representations based on a CQT (CQTSG and ZPCP) seem to outperform the filterbank-based representations (FBSG and MPCP). This can be explained by the smaller bandwidth of the used filters, which can overly penalize pitch imprecisions. Another possible reason is the noise level in the low frequencies, which can be very high when a bass drum is present. Thus, a good solution is sometimes to completely discard very low frequencies, which is the case in our CQT.

VI. Discriminative Learning

In this section, we expose another strategy for the discriminative learning of the mapping matrix, thanks to a Conditional Random Fields (CRF) model [18]. In this method, the alignment model is taken into account in the learning process.

A. Markovian Conditional Random Fields (MCRF) Model

The alignment strategy of (13) can be derived from a special case of a Markovian Conditional Random Fields (MCRF) model as presented in [17]. The MCRF model is a discriminative probabilistic model which allows for the calculation of any concurrency sequence probability $\underline{c}_{1:N}$, given a sequence of observation vectors $\underline{v}_{1:N}$. In order to clarify the presentation, the boundary indices $_{1:N}$ will be omitted in the following when no ambiguity is introduced. The probability of the concurrency sequence is given by:

$$P(\underline{c}|\underline{v}) = \frac{1}{Z(\underline{v})}\phi(c_1, v_1)\prod_{n=2}^{N}\psi(c_n, c_{n-1})\phi(c_n, v_n), \tag{14}$$

where $Z(\underline{v})$ is a normalization factor and ψ and ϕ are non-negative potential functions. The observation function ϕ is here defined by

$$\phi(c_n, v_n) = \exp\left\{ -\mu f\left(c_n, v_n\right)\right\}, \tag{15}$$

where μ is a positive weight parameter. The transition function ψ used here only constrains $\underline{c}$ to be an acceptable concurrencies sequence, i.e. to follow the score order[3]. With these definitions, the value of μ does not influence the decoding of the model and the most probable concurrency sequence is given by (13).

B. Maximum Likelihood (ML) Criterion

Since the MCRF is a probabilistic model, a natural framework for learning the parameters is to employ the Maximum Likelihood (ML) criterion, i.e. to maximize the probability of the ground truth concurrency sequences. We write $\Theta = (\mu, \mathbf{W})$ for the parameters of the model. The value of μ is learned as well as $\mathbf{W}$, since it has an influence on the probabilities. The optimal parameters are then defined by:

$$\hat{\Theta}^{ML} = \underset{\Theta}{\operatorname{argmax}} \prod_{s} P\left(\underline{c}^s \middle| \underline{v}^s; \Theta\right) \tag{16}$$

where $P(\cdot|\underline{v};\Theta)$ denotes the probability given by the model with parameters Θ. We define:

$$F_1(\underline{c}_{1:N}, \underline{v}_{1:N}) = -\sum_{n=1}^{N} f\left(c_n, v_n\right). \tag{17}$$

Then, the log-likelihood can be written:

$$\mathcal{L}(\Theta) = \sum_{s}\left\{\mu F_1\left(\underline{c}^s, \underline{v}^s\right) - \log Z\left(\underline{v}^s\right)\right\}. \tag{18}$$

This function is concave with respect to μ and the corresponding derivative is

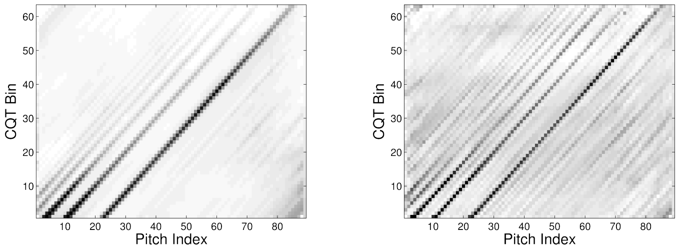
$$\frac{\partial\mathcal{L}(\Theta)}{\partial\mu} = \sum_{s}\left\{F_1\left(\underline{c}^s, \underline{v}^s\right) - E\left[F_1\left(\underline{C}^s, \underline{v}^s\right)\middle|\underline{v}^s; \Theta\right]\right\}, \tag{19}$$

where $\underline{C}^s$ is a random variable representing the concurrency sequence of the s-th training sample and $E[\cdot|\underline{v}^s;\Theta]$ denotes the expectation with respect to the conditional distribution $P\left(\underline{C}^s\middle|\underline{v}^s;\Theta\right)$. The expectation term can efficiently be computed thanks to a variant of the forward-backward algorithm [37], making it tractable to calculate this derivative.

Unfortunately, the log-likelihood is not concave with respect to $\mathbf{W}$. Nevertheless, the gradient can be expressed in a relatively simple form:

$$\frac{\partial\mathcal{L}(\Theta)}{\partial\mathbf{W}_{i,j}} = \mu\sum_{s}\left\{\frac{\partial F_1}{\partial\mathbf{W}_{i,j}}\left(\underline{c}^s, \underline{v}^s\right) - E\left[\frac{\partial F_1}{\partial\mathbf{W}_{i,j}}\left(\underline{C}^s, \underline{v}^s\right)\middle|\underline{v}^s; \Theta\right]\right\}. \tag{20}$$

[3] The expression of ψ is: $\psi(c_n, c_{n-1}) = \mathbf{I}\{c_{n-1} - c_n \in \{0, 1\}\}$ with $\mathbf{I}$ the indicator function.



(a) Minimum Divergence (MD) criterion

(b) Maximum Likelihood (ML) criterion

Figure 7. Comparison of the mapping matrices learned by our two criteria, for the CQTSG representation with KLs distance. The gray scale is the same on both images.

Some preliminary experiments using a Limited memory BFGS (L-BFGS) algorithm [38] did not prove very conclusive. Therefore, we resort to a simple algorithm where μ and $\mathbf{W}$ are alternatively updated in the direction of their gradient, with an adaptive step. Because of the complexity of the gradient calculation, we limit the iteration number to 100. The initialization is done with the result of the MD learning, based on the intuition that this value is close to the optimum. We also suppose that this initialization does not overly favor the ML learning strategy compared with MD, since the optimization criteria are different.

C. Results

The experiments are conducted with all the representations presented in Section III-D. However, we only exploit here the KLs distance, since it proves to be the most efficient whatever the representation. The learning process is run on the same learning database as in the previous experiment.

Figure 7 compares the mapping matrices learned with both criteria. We can observe that the ML strategy yields a smoother distribution of the 'energy' along the observation bins, with fewer small values. This can be explained by the notion of maximum entropy*, on which CRFs are based [37]. Indeed, the ML learning does not aim at fitting the observations, but rather at discriminating the concurrencies. Intuitively, extreme values are then given only to the bins which are really useful for the discrimination of the concurrencies, and the other bins are given 'medium' values.*

Alignment experiments using the same approach as in Section V-C2 are also run. Since the ML learning strategy takes into account the alignment model, one could expect an increase of the obtained precision. However, whereas the PS and FBSG representations are further improved compared to the use of the MD learning, the results of the other representations are dramatically reduced. For example, the alignment rate of the CQTSG semigram drops from 58.2% to 10.4% on the RWC corpus. A reason for this is the fact that the used strategy maximizes the probability of the ground-truth sequence, but does not limit the probability of the other sequences, which may also benefit from the optimization. Thus, nothing ensures that the ground-truth will be the most probable sequence and the alignment rates are not guaranteed to increase, even on the training set.

Mapping	PS	CQTSG	FBSG	ZPCP	MPCP
H	66.3	64.9	60.4	56.9	52.4
MD	69.9	68.2	61.7	58.6	54.6
ML	70.3	71.0	63.8	57.6	52.3

Table III
RECOGNITION RATES OBTAINED WITH THE BASIC MODEL, FOR THE KLS DISTANCE.

By a precise examination of the results, we noticed a number of aberrant alignments, where most of the piece was decoded as the 'silence/noise' state (which is present at the beginning and the end of each score). The feature function indeed introduces a bias toward this state. This is due to the form of the optimized likelihood. The partial derivative of (20) can be developed as:

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{W}_{i,j}} = \mu \frac{\partial}{\partial \mathbf{W}_{i,j}} \left(\sum_{s} \sum_{n=1}^{N_s} \left\{ E\left[f\left(C_n, v_n^s\right) \middle| \underline{v}^s ; \Theta' \right] - f\left(c_n^s, v_n^s\right) \right\} \right) \Bigg|_{\Theta'=\Theta} . \tag{21}$$

Hence, the ML learning strategy aims at maximizing the difference between the feature function of the ground-truth label and the expectation of this feature function (computed over all the possible labels). While intuitive, this process can lead to a specific issue. Indeed, a label probability is the sum of the probabilities of all the sequences containing this label. Thus, more emphasis can be put on a label enclosing many sequences of moderate probability than on a label which is contained in an isolated high-probability sequence. Hence, in our alignment model, some labels 'far' from the ground truth path are given little importance in the discriminative learning. This is the case for the 'silence' states which receive very low weights, resulting in an overestimation of the corresponding feature function. These labels can then accept virtually any observation. On the other hand, the templates of the pitched concurrencies are trained to 'reject' the other probable labels. Hence, the corresponding feature functions are much more selective, and thus more sensitive to noise or pitch imprecision.

In order to overcome this problem, we adopt an ad hoc *strategy which modifies the feature function of the 'silence' labels in the decoding phase, so that it has the same order of magnitude as for the other labels. The value is not calculated using the corresponding template, but as the mean of the feature function of the 10 surrounding concurrencies in the score. The alignment results are displayed in Table III and compared to the other learning strategies. It must be noted that the modification of the feature function has also been applied to the other mapping matrices, without introducing a significant variation of the results. One can observe that the ML criterion allows for an improvement of the alignment rates for the spectrogram and semigram representations. The chromagrams, however, do not benefit from this learning approach, probably because of a more limited 'discriminative power' due to their smaller dimension. The results are nevertheless at least as good as with the heuristic mapping.*

Mapping	PS	CQTSG	FBSG	ZPCP	MPCP
H	73.9	73.6	72.0	67.5	64.5
MD	76.4	75.6	72.5	69.3	65.3
ML	76.7	77.5	73.5	68.9	64.9

Table IV
RECOGNITION RATES OBTAINED WITH THE MCRF MODEL WITH ONSET FEATURE, FOR THE KLs DISTANCE.

Learning	PS	CQTSG	ZPCP
H	96.8	96.9	95.9
MD	96.7	97.0	95.8
ML	96.9	97.7	96.0

Table V
RECOGNITION RATES OBTAINED WITH THE HTCRF MODEL WITH KLs DISTANCE.

VII. Influence of the Learned Mappings on State-of-the-Art Alignment Systems

We now evaluate the influence of the projection learning on the accuracy of two alignment systems which exploit additional pieces of information to the pitch observation vectors. These systems have been presented in details in [17].

A. Introduction of an Onset Feature

The first system tested is the simplest system of [17]. It is a Markovian CRF using an additional onset feature for discriminating between the attack *and* sustain *phases of each concurrency. Similar to* (15)*, the weight given to this onset feature is controlled by a parameter denoted by ν.*

For this experiment, μ is learned by the Maximum Likelihood criterion presented in Section VI-B. The same strategy has been attempted for learning the optimal value of ν. However, as already mentioned, ML learning does not necessarily lead to optimal parameter values, in the sense of the alignment rate. We then resort to a coarse grid search in order to adjust this parameter, in the same way as in Section IV-D.

The obtained results are presented in Table IV. The introduction of the onset feature allows for a significant improvement (at least +6% absolute) in the accuracy of all the tested systems, whereas the ranking of the representations does not change. One can also observe the advantage of the learning of the mapping matrix $\mathbf{W}$*. Indeed, for every representation, both learning strategies outperform the heuristic mappings.*

B. Alignment with the Hidden Tempo CRF Model

In a final set of experiments, we employ the Hidden Tempo CRF (HTCRF) model exposed in [17]. The HTCRF extends the Markovian CRF used previously by incorporating an explicit and very precise temporal model. Hence, the potential function ψ of (14)*, controlling the label transitions, depends on the concurrency durations. Furthermore, the value of this potential function also depends on an additional hidden variable representing the current* tempo *of the piece. In our experiment, we only use the best representation of each type, namely the power spectrum, the CQTSG semigram and the ZPCP chromagram. The three considered mappings are tested.*

The HTCRF model requires a discrete set $\mathcal{T}$ of possible tempi. The values used here are, in beat per minute:

$$\begin{aligned}\mathcal{T} =\{&28, 30, 34, 40, 48, 56, 64, 72, 80, 88, 96, 104,\\ &112, 120, 132, 146, 160, 176, 192, 208, 224, 240\}.\end{aligned} \tag{22}$$

Due to the high complexity of the HTCRF model, a learning of all the parameters with the ML criterion is not possible. Thus, we use the values estimated in the previous experiment (Section VII-A) with the MCRF model for μ and ν, and the other parameters have been set through a grid search strategy. In these experiments, the modified version of the 'silence' feature proved slightly more effective than the original one, for both ML and MD learning strategies. Therefore, the results presented in Table V correspond to these settings.

As expected, the introduction of a precise temporal model greatly improves the performance of all the tested systems. The obtained alignments are then very accurate, since more than 95% of the onsets are correctly recognized within a 100 ms tolerance threshold, for all the tested systems. Since the HTCRF system adds strong constraints on the alignments, the differences between the tested systems are smaller than with the MCRF model. Nevertheless, the impact of the learning is visible on the CQTSG representation, where the ML strategy allows for a significant improvement of the alignment rate compared to the heuristic mapping (97.7% against 96.9%).

VIII. Conclusion and Perspectives

In this paper, we have described a template-based feature function for the matching of a symbolic and an audio representation of a musical piece. We have proposed two strategies for the learning of the mapping from the symbolic to the observation domain, when it can be written as a linear transformation. The evaluations, performed on a large database of polyphonic music, indicate that this learning can lead to a significant increase of the precision of several CRF alignment systems. The results also show that in many cases, the minimum divergence *learning criterion leads to a good alignment accuracy. However, with the most complex CRF model, the highest performance is obtained using the Maximum Likelihood (ML) criterion, indicating that this discriminative learning algorithm has the potential to improve the matching of the symbolic elements, at a fine precision level.*

Furthermore, we have compared several representations of the audio performance as well as several distance functions, for this alignment task. Our results indicate that the symmetric Kullback-Leibler divergence is a good choice of distance, and that both the spectrogram and the CQT-based semigram representations provide very accurate alignments.

Many perspectives can be imagined for the continuation of this work. First, one can investigate the use of other kinds of features, through other distance functions, but also different observations, e.g. *the output of a Non-negative Matrix Factorization algorithm such as in [22]. In this work, we have only investigated the exploitation of a single "pitch feature function". Nevertheless, the CRF framework allows for any number of features. In fact, the symmetric Kullback-Leibler divergence used here is already constructed as the* *superposition of two distance functions. In the same way, any mixture of features can be imagined, whose respective weights can be learned. Features related to other points than the current frame could also be exploited, such as in [17].*

Some results show that the ML learning criterion does not always yield the best alignment accuracy. Thus, other criteria could be investigated, both for the learning of the features and for the decoding of the label sequence, such as the minimum segmentation error *proposed in [25]. Finally, the use of a single template for each pitch in the construction of all the concurrency template is a rather strong constraint. Indeed, it disregards the possibly large variations due to different instruments and recording conditions. Hence, one can imagine an adaptive approach which would adjust the projection matrix to the characteristics of each piece, and possibly to each of the present instruments.*

Acknowledgment

The authors would like to thank Dr. Angélique Drémeau for her valuable comments and advices in the writing of this paper.

References

[1] S. Dixon and G. Widmer, "Match: a music alignment tool chest," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2005, pp. 492–497.

[2] 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.

[3] J.-L. Durrieu, G. Richard, B. David, and C. Fevotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," IEEE Trans. Audio, Speech, Language Processing, vol. 18, no. 3, pp. 564 –575, 2010.

[4] L. Oudre, "Reconnaissance d'accords à partir de signaux audio par l'utilisation de gabarits théoriques / template-based chord recognition from audio signals," Ph.D. dissertation, TELECOM ParisTech, Nov. 2010.

[5] N. Orio, S. Lemouton, and D. Schwarz, "Score following: State of the art and new developments," in Proc. of New Interfaces for Musical Expression Conf., 2003, pp. 36–41.

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

[7] C. Raphael, "A probabilistic expert system for automatic musical accompaniment," Journal of Computational and Graphical Statistics, vol. 10, no. 3, pp. 487–512, 2001.

[8] D. Damm, C. Fremerey, F. Kurth, M. Müller, and M. Clausen, "Multimodal presentation and browsing of music," in Proc. of Inter. Conf. on Multimodal Interfaces, 2008, pp. 205–208.

[9] N. Orio, "A system for the automatic identification of musicworks," in Proc. of IEEE Inter. Conf. on Image Analysis and Processing - Workshops, 2007, pp. 15–20.

[10] Y. Han and C. Raphael, "Informed source separation of orchestra and soloist," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), Utrecht, the Netherlands, Aug. 2010, pp. 315–320.

[11] A. Liutkus, R. Badeau, and G. Richard, "Informed source separation using latent components," in LVA/ICA, Saint Malo, France, Sep. 2010, pp. 498–505.

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

[13] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in Proc. of IEEE Workshop Applicat. of Signal Processing to Audio and Acoust (WASPAA), 2003, pp. 185–188.

[14] N. Montecchio and A. Cont, "A unified approach to real time autio-to-score and audio-to-audio alignment using sequential monte-carlo inference techniques," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), May 2011, pp. 193–196.

[15] O. İzmirli and R. Dannenberg, "Understanding features and distance functions for music sequence alignment," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2010, pp. 411–416.

[16] C. Joder, S. Essid, and G. Richard, "Optimizing the mapping from a symbolic to an audio representation for music-to-score alignment," in Proc. of IEEE Workshop Applicat. of Signal Processing to Audio and Acoust (WASPAA), New Paltz, NY, USA, Oct. 2011.

[17] ——, "A conditional random field framework for robust and scalable audio-to-score matching," IEEE Trans. Audio, Speech, Language Processing, vol. 19, no. 8, pp. 2385–2397, Nov. 2011.

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." in Proc. of Inter. Conf. Machine Learning, 2001, pp. 282–289.

[19] R. B. Dannenberg and H. Mukaino, "New techniques for enhanced quality of computer accompaniment," in Proc. of Inter. Computer Music Conf. (ICMC), 1988, pp. 279–289.

[20] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), 2009, pp. 1869–1872.

[21] B. Niedermayer and G. Widmer, "A multi-pass algorithm for accurate audio-to-score alignment," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2010, pp. 417–422.

[22] A. Cont, "Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), 2006, pp. 245–248.

[23] N. Montecchio and N. Orio, "A discrete filterbank approach to audio to score matching for score following," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2009, pp. 495–500.

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

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

[26] A. Cont, D. Schwarz, and N. Schnell, "Training ircam's score follower," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), vol. 3, 2005, pp. 253–256.

[27] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in Proc. of Inter. Computer Music Conf. (ICMC), 2003, pp. 27–33.

[28] P. Bellini, I. Bruno, and P. Nesi, "Assessing optical music recognition tools," Computer Music Journal, vol. 31, pp. 68–93, Mar. 2007.

[29] T. Virtanen, A. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), Apr. 2008, pp. 1825 –1828.

[30] C. Joder, S. Essid, and G. Richard, "A comparative study of tonal acoustic features for a symbolic level music-to-score alignment," in Proc. of IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP), 2010, pp. 409–412.

[31] F. Soulez, X. Rodet, and D. Schwarz, "Improving polyphonic and poly-instrumental music to score alignment," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2003, pp. 143–148.

[32] M. Müller, Information Retrieval for Music and Motion. Springer Verlag, 2007.

[33] 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.

[34] M. Müller, F. Kurth, and T. Röder, "Towards an efficient algorithm for automatic score-to-audio synchronization," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2004, pp. 365–372.

[35] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," SIAM Journal on Scientific Computing, vol. 21, no. 1, pp. 1–23, 1999.

[36] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in Proc. of Inter. Soc. for Music Information Retrieval Conf. (ISMIR), 2002, pp. 287–288.

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

[38] J. Nocedal, "Updating quasi-newton matrices with limited storage," Mathematics of Computation, vol. 35, no. 151, pp. 773–782, 1980.

.

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 completed his Ph.D thesis on music signal processing at the department of Signal and Image Processing at Telecom ParisTech in 2011. Then, he worked at the Technische Universität München, Munich, Germany, on machine learning for speech and music processing. In February 2013, he joined the European Patent Office in Munich as a patent examiner.

Slim Essid is an Associate Professor at the Department of Image and Signal Processing of Telecom ParisTech with the Audio & Waves group. He received the state engineering degree from the École Nationale d'Ingénieurs de Tunis 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), in 2005, after completing a thesis on automatic audio classification. His research interests are in machine learning for multimodal signal analysis with applications to music information retrieval, audiovisual content analysis, and human behavior and activity analysis. He has published over 60 peer-reviewed conference and journal papers with more than 50 distinct co-authors. He has been involved in various French and European research projects among which are Quaero, Infom@gic, Networks of Excellence Kspace and 3DLife, and collaborative projects REVERIE and VERVE. He serves on a regular basis as a reviewer for various audio and multimedia conferences and journals, for instance various IEEE transactions, and as an expert for research funding agencies. More information on http://www.telecom-paristech.fr/ essid.

Gaël Richard (SM'06) 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 150 papers and inventor in a number of patents and is also one of the experts of the European commission in the field of audio signal processing and man/machine interfaces. He was an Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing between 1997 and 2011 and one of the guest editors of the special issue on "Music Signal Processing" of IEEE Journal on Selected Topics in Signal Processing (2011). He currently is a member of the IEEE Audio and Acoustic Signal Processing Technical Committee, member of the EURASIP and AES and senior member of the IEEE.