# A COMPARATIVE STUDY OF TONAL ACOUSTIC FEATURES FOR A SYMBOLIC LEVEL MUSIC-TO-SCORE ALIGNMENT

*Cyril Joder, Slim Essid, Gaël Richard*

Institut Télécom - Télécom ParisTech - CNRS/LTCI
37 rue Dareau, 75014 Paris, France

**ABSTRACT**

In this paper we review the acoustic features used for music-to-score alignment and study their influence on the performance in a challenging alignment task, where the audio data is polyphonic and may contain percussion. Furthermore, as we aim at using "real world" scores, we follow an approach which does exploit the rhythm information (considered unreliable) and test its robustness to score errors.

We use a unified framework to handle different state-of-the-art features, and propose a simple way to exploit either a model of the feature values, or an audio synthesis of a musical score, in an audio-to-score alignment system. We confirm that *chroma vectors* drawn from representations using a logarithmic frequency scale are the most efficient features, and lead to a good precision, even with a simple alignment strategy. Robustness tests also show that the relative performance of the features do not depend on possible musical score degradations.

***Index Terms***— music information retrieval, automatic alignment, acoustic features

## 1. INTRODUCTION

Audio-to-score alignment is the task of synchronizing a musical score with a its audio performance. The result is a mapping between each instant in the recording and a position in the score. This task has been extensively studied from the *real-time* point of view, for the application of automatic accompaniment of a musician [1, 2]. In this case, the problem is also known as *score following*.

We deal here with the *offline* alignment (or audio synchronization) of an audio recording with its score. Indeed, a temporal alignment can be useful in applications where the real-time constraints do not apply, such as score-controlled audio browsing or automatic indexing, in order to take advantage of the numerous scores than can be found freely, on the internet for instance. However, these scores are often error-prone. Hence, as the indicated rhythm may be unreliable, we choose not to use the note duration information. We are interested in an alignment at a *symbolic* level, i.e. in which the result is the time indexes of the score notes or chords.

An alignment system can be separated into two layers. A low-level layer extracts features describing the instantaneous content of the audio signal. These features are used by a high-level layer, which performs the alignment thanks to a model of the temporal evolution of the music. Most works on audio-to-score alignment and score following use specific low-level features along with their own high-level system. The focus is usually put on the temporal model [3, 4, 1] and the low-level layer is often introduced with little discussion about its efficiency. To the authors' knowledge, there has been no comprehensive study on the relative performance of the different features proposed for this task in previous works. Furthermore, although polyphonic music has already been used to assess the precision of some alignment systems (see for example [5, 4, 6]), large scale comparative evaluations on the same database at the symbolic level, such as the last MIREX campaign on this domain[7], have been almost exclusively performed with monophonic or slightly polyphonic classical music.

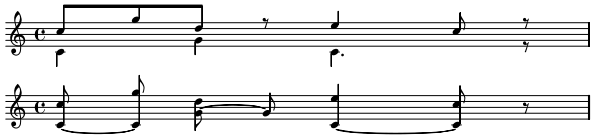

**Fig. 1**. An example of the score representation. Top: a "normal" musical score; bottom: the corresponding chord representation

In the present work, we study the specific influence of the low-level layer on an audio-to-score alignment system in a challenging polyphonic case. We use a unified framework to compare different models, including a novel simple method to perform an alignment at the symbolic level using an audio synthesis of the score. We also test the behavior of the models when confronted to errors in the score and find that the relative performances are not affected by these errors.

The rest of this paper is organized as follows: in Section 2, we define the alignment problem and separate the low-level layer from the temporal model. The low-level models considered in the study are detailed in Section 3. We then expose in Section 4 the results of our experiments on the alignment performance induced by the different features. Finally, conclusions are provided in Section 5.

## 2. THE ALIGNMENT SYSTEM

From a general point of view, a musical score is a list of notes described by their pitch, onset and offset times. However, in order to locate positions in a polyphonic score, it is useful to have a *fully ordered* representation of this score. As in [4] we consider a score as a sequence of *chords*, which are sets of notes that sound together, indexed by their onset times. Figure 1 represents the conversion from a "normal" score to our chord sequence representation.

The output of an alignment algorithm is the sequence of chords which "best match" the audio signal. Let $\mathbf{y}=y_1,\dots,y_n$ be the feature sequence extracted from the signal. If $S_t$ is a random variable describing the current chord at time $t$, the low-level layer calculates the *local likelihood* $p(y_t|S_t=s)$ of each chord $s$ corresponding to the observation $y_t$. The high-level layer then determines the optimal (in some sense) sequence of chords given the sequence of feature observations. The high-level module used in this work is very simple, as it searches for the *maximum likelihood* path $\hat{\mathbf{S}}$, defined by :

$$\hat{\mathbf{S}} = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}}\, L\big(\mathbf{Y}\big|\mathbf{S}\big) = \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmax}} \prod_{i=1}^{n} p(Y_i|S_i),$$

where $\mathcal{S}$ is the set of acceptable paths. These acceptable paths are all the paths which begin with the first chord and end with the last one of the score, with no "heaps" (no chord may be skipped). This optimum can be efficiently found thanks to the Viterbi algorithm.

It is important to note that our alignment strategy does not take into account chord durations indicated in the score. This permits to integrate the fact that two interpretations of the same piece can face very large deviation in note durations while the same sequence of chords will be observed. Thus, the only temporal information that is used is the order of the chords.

## 3. AUDIO FEATURES FOR ALIGNMENT

The low-level layer estimates the likelihood of each chord for each frame of the audio recording. Since a chord is a set of *pitched* notes, we use features which describe the spectral content of the audio. In order to evaluate the impact of the low-level layer, we compare several feature models, which can be divided into three broad classes. Table 1 sums up the representations which are studied in this work.

### 3.1. Spectral models

The power spectrum drawn from a Short-Term Fourier Transform (STFT) of the audio signal is used in many score following works [2, 4, 1], because of the low complexity of this transform. In order to calculate the likelihood of a power spectrum $\{y(\omega)\}$ (where $\omega$ spans the frequency bins) given a chord $s$ containing $|s|$ notes with fundamental frequencies $\{f_1, \ldots, f_{|s|}\}$, two methods have been tested.

**Generative Spectrum.** The first method, drawn from [4], uses a template model of the power spectrum, given the notes of the chord. For one note of fundamental frequency $f$, the model $g_f(\omega)$ is a mixture of Gaussians, where the means correspond to the harmonics. We consider 5 components, whose standard deviations are set to one semitone and whose weights are quadratically decreasing.

Then, the corresponding chord model $g$ is a mix of the one-note templates, with an additional noise term:

$$g(\omega) = \frac{1-q}{|s|} \sum_{n=1}^{|s|} g_{f_n}(\omega) + q\, U_{[0,\omega_{\max}]}(\omega). \qquad (1)$$

Here, $U_{[0,\omega_{\max}]}(\cdot)$ is the uniform pdf, and the parameter $q$ controls the amount of "noise" in the model. In the experiments, ten different values have been tested between 0 and 0.95. For silent chord (where $s$ is the empty set $\emptyset$), only the noise term remains ($q$=1). This model is then used as a probability distribution over the frequency bins.
According to Raphael's Generative Spectrum (RGS) model [4], the likelihood is estimated by the formula:

$$p_{RGS}(y|S=s) = C(y) \prod_{\omega} g(\omega)^{y(\omega)}, \qquad (2)$$

which calculates the likelihood of $y$ if it is seen as a histogram of random samples from the distribution $g$. This model is referred to as *histogram model*. In our application, the value of $C$ has no impact on the alignment results, since it is the same for every path.

Cont calculates the chord likelihood [1] thanks to a probabilistic measure of a normalized version $\bar{y}$ of the power spectrum: $p_{CGS}(y|S=s) = \exp\big(-D(\bar{y}||g)\big)$, where $D(\cdot||\cdot)$ denotes the Kullback-Leibler divergence. The exponential function is used to convert the divergence into a probability estimate.

**Peak Spectral Match.** We also estimate the chord likelihoods by the value of the Peak Spectral Match (*PSM*) exposed in [2], which is the ratio of the signal energy in the expected frequency bands, over total energy. In the special case of silence ($s=\emptyset$), the value of the likelihood is given by : $p_{PSM}(y|S=\emptyset) = (\frac{\min(-E_{\text{dB}},\gamma)}{\gamma})^{\nu}$, where $E_{\text{dB}}$ is the normalized energy of the frame, in dB (its maximum is 0). We set the scale parameter $\nu$ is to 5 and the threshold $\gamma$ to 30 dB.

### 3.2. Semitone Energy Features

We also test spectral representations which follow the same scale as the musical score. The *semitone* features represent the spectral power in the musical chromatic scale ($C_1$,$C\#_1$,...).

We compute such features thanks to the same bank of elliptic filters as in [3]. We also use a constant Q transform (CQT), with a quality factor set to one semitone. These features are respectively denoted by *FBSE* (for FilterBank Semitone Energy) and *CQTSE*.

Three different approaches are then used to calculate the likelihood of a chord. The first estimate is, as for the PSM (Sec. 3.1), the ratio of the power in the expected semitone bands, over the total power. We call it the *ratio* method. As in [3], the spectral power located at the first two harmonics of the note is also taken into account.

For the two other methods, a template vector $g$ is built for each chord, as the superposition of note templates and of a noise component. The template values are 1/3 in the bins of the first three note partials, and 0 elsewhere. The likelihood is then estimated by :
– the *histogram model* of (2): $p_{\text{hist}}(y|g) = \prod_{\omega} g(\omega)^{y(\omega)}$, where $y$ is the observation vector;
– the value of the *cosine measure*: $p_{\cos}(y|g) = \frac{\langle y\cdot g\rangle}{\|y\|\|g\|}$, where $\langle\cdot\rangle$ denotes the inner product and $\|\cdot\|$ is the $L^2$ norm.

### 3.3. Pitch Class Profile Representations

We also consider *pitch class profiles* (*PCP*), or *chroma vector* representations, which consist in a 12-component vector corresponding to the spectral energies of the 12 musical pitch classes (A, A#,...). These representations have been shown to perform well in the task of audio-to-audio alignment [8, 9].

Many ways have been proposed to calculate such features. We compare here four different algorithms to obtain chroma vectors.

– The first representation is the integration of the FBSE features over the different octaves. This feature is denoted by FBPCP.

– We also use an algorithm proposed by Peeters [10], which sums the STFT magnitude for each pitch class. This representation is denoted by PPCP (for Peeters' *PCP*).

– A third chroma vector (*ZPCP*) is calculated according to Zhu's method [11], which performs a peak-picking on the CQT, and then sums the amplitude corresponding to all the octaves.

– The last representation is Gómez's Harmonic Pitch Class Profile (*HPCP*) [12]. Its calculation involves peak-picking on the STFT magnitude with quadratic interpolation, and the integration of the energy of the harmonics for each chromatic bin.

The likelihoods are then estimated by the same methods as in Section 3.2: *ratio*, *histogram* and *cosine* measures. The latter two methods need templates corresponding to the chords. Following [8], the template is created based on the notes which are present in the chord. For example, a chord containing notes $C_3$, $E_3$, $G_3$ and $C_4$ leads to the template $(0,0,0,2,0,0,0,1,0,0,1,0)$, where the vector components correspond to pitch classes from A to G#. A noise component is added to these templates, as in Eq. (1).

| Acron. | Meaning | Acron. | Meaning |
|---|---|---|---|
| *RGS* | Raphael's Generative Spec. | *HPCP* | Harm. Pitch Class Profile |
| *CGS* | Cont's Generative Spec. | *ZPCP* | Zhu's PCP |
| *PSM* | Peak Spectral Match | *PPCP* | Peeters' PCP |
| *FBSE* | FilterBank Semitone Energy | *FBPCP* | FilterBank PCP |
| *CQTSE* | CQT Semitone Energy | | |

**Table 1**. Summary of the low-level features tested.

### 3.4. Synthetic template-based likelihood computation (*syntemp*)

For the chord likelihood estimation, we test an additional method that uses an audio synthesis of the score (thanks to the TiMidity++ software[1]), in order to obtain more realistic chord templates (which are referred to as *syntemp*), as an alternative to the simple theoretical templates presented in Section 3. For a specific chord $s$ in the score, let $n_1, \ldots, n_L$ be the indexes of the $L$ frames where the chord $s$ is playing in the synthesis. Let $\hat{g}_{n_1}, \ldots, \hat{g}_{n_L}$ be the feature vectors observed on these frames. We estimate the likelihood of $s$ by $p_{\text{synth}}(y|s) = \max_{\ell \in \{1,\ldots,L\}} p_{\cos}(y|\hat{g}_{n_\ell})$, where $y$ is the observation. This approach computes a single likelihood value for each chord, despite the variable durations of the chords in the synthesized data.

## 4. EXPERIMENTS

### 4.1. Database and Experimental Settings

To evaluate the alignment systems, we need ground-truth MIDI files which are perfectly aligned to the audio signals. We first exploit the MIREX'06 score following evaluation database, comprising four pieces of classical monophonic (or slightly polyphonic) mono-instrumental music. We also use 93 pop songs from the RWC database [13]. These songs are polyphonic multi-instrumental pieces, most of which contain percussion. In order to simulate the case where the scores are unreliable, we simply use the ground-truth MIDI files and, depending on the experiments, we choose to exploit only certain pieces of information. Here, we discard duration information to evaluate the capacity of the system to infer this information based on the resulting sequence of chord labels.

In our experiments, the audio signals are converted to mono and downsampled to 16 kHz. For signal analysis, we use 50 ms frames with a 20 ms hop-size, and 2048 frequency bins for the STFT. For each feature, ten values of the noise parameter $q$ (see eq.1) are evaluated, from the set $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. The figures given below correspond to the best value.

The chosen evaluation measure gives the error rate, that is the proportion of onsets which are detected beyond a threshold $\theta = 300$ ms around the real onset time. This threshold is chosen after the MIREX'06 contest.

### 4.2. Results and Discussion

The most important results of the experiments on the monophonic and polyphonic databases are presented in Figure 2.

The first tests are run on the MIREX database (Figure 2, up). In this contest, the best system obtained a $9.9\%$ error rate. Apart from the PPCP system, which obtains a very large error rate ($26.0\%$), all our system perform better. This is not really surprising since, although we do not use temporal information, we do not apply real-time (nor even causal) constraints either.

**Monophonic vs. Polyphonic Music.** Consistently with one's intuition, the error rates are much higher on polyphonic than on monophonic music, since this data is more complex. Moreover, whereas the three features classes (power spectrum, semitone and *PCP*) can obtain similar results on the monophonic database (five of the feature models obtain between 3% and 5.3% error rates), the spectral models perform much worse than the others on the polyphonic case (+25% error rate at least). These models appear to be unadapted to polyphonic music, as they are too much dependant on the timbre and too sensitive to background noise. Thus, the choice of the feature model is all the more important as the data is complex.

[1]TiMidity++: `http://timidity.sourceforge.net/`

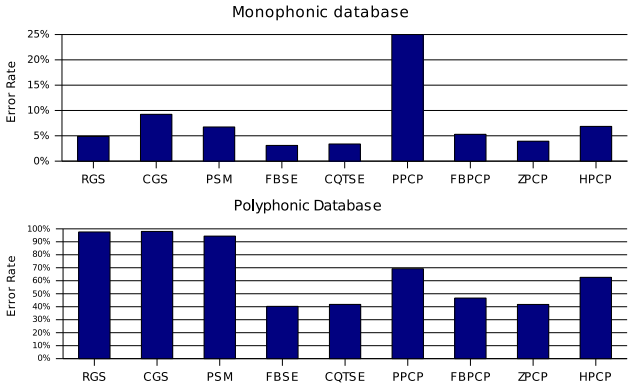


**Fig. 2**. Results of the theoretical models. Scores are given for the best parameter settings.

**Limitations of the STFT.** On the polyphonic database, the five worse systems are all those which exploit a STFT. This brings to light the problem of an insufficient resolution of this transform in low frequencies. Whereas this limitation can be overcome in the monophonic case (by taking into account neighboring bins as in the spectral models), the presence of several close partials in polyphonic music makes it preferable to use a logarithmic frequency scale.

**Semitone vs. *PCP* Features.** The four features which exploit such a scale (*FBSE*, *CQTSE*, *FBPCP*, *ZPCP*) obtain comparable scores, between 40.3% and 46.7% on the real polyphonic data (Figure 2, bottom), with their best parameter sets. Chroma features have already been proven successful for audio synchronization [8]. Here we empirically show that the octave information does not improve the alignment results.

**Likelihood Calculation Method.** On the other hand, the likelihood calculation method can have an influence. For example, the error rates of the *CQTSE* features with the *ratio*, *cosine* and *histogram* measures are respectively $61.7\%$, $47.0\%$, and $41.8\%$.

The *histogram* and *cosine* measures obtain similar results. However, the former is generally more efficient because the factor $g^y$ in (2) penalizes audio events (notes) which are not expected in the score (small $g$ and large $y$), resulting in a good discrimination between different chords. The *ratio* method performs poorly, because of its bias towards "abundant" chords (for a chord containing all the possible notes, all the frequency bins are expected, resulting in an energy ratio of 1) and its inability to take into account a model of noise.

**Theoretical templates vs. *syntemp*.** As shown in Fig. 3, the behavior of the *syntemp* systems are similar to their "theoretical templates" counterparts. This indicates that our theoretical models are quite well suited for this application, and that the observed tendencies are directly linked to the features rather than the likelihood calculation method. For almost every feature, the use of *syntemp* improves the system performance, as the synthesis allows for more re-

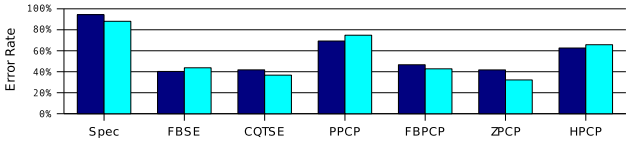


**Fig. 3**. Comparison of theoretical templates (dark) and *syntemp* (light) on the real polyphonic database.

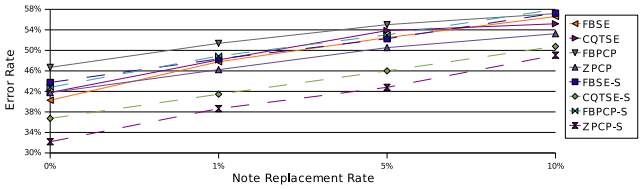


**Fig. 4**. Error rates of several systems as a function of the note replacement rate ("-S" stands for with *syntemp*)

alistic templates. It may also be explained by a possible "temporal bias": as the score rhythm is used for the synthesis, the number of synthetic templates for a chord depends on its score duration. Hence, longer chords may be favored by a greater number of *syntemps*.

The result degradation of *FBSE* can be explained by timbre differences, which modify the relative weights of the partials, and tuning or inharmonicity problems, which can affect our filterbank representations, since these filters are narrower than those of the CQT. This latter problems explain the poor results of *PPCP*, because of the sensitivity of the STFT-to-chroma mapping in low frequencies.

**Robustness of features to presence of percussion.** In the polyphonic database, 9 songs contain no percussion. On these particular parts of the data, the performances are better. For example, the error rates of *CQTSE* and *ZPCP* are respectively 34.3% and 32.8%, whereas they are both 41.8% on the whole database. Hence, percussion does affect the performance of these pitch-based features. However, the ranking is not modified, indicating that the best features can be chosen regardless of the music instrumentation.

**Robustness to error-prone scores.** Experiments are run on the polyphonic real audio files whose scores have been modified. We considered three types of modification: suppression or pitch change on randomly chosen notes; and addition of notes (with random pitch, time and duration). Three modification ratios are considered: 1%, 5% and 10%. The error rates are then calculated only on the unmodified chords. Here are tested only the most effective features. Figure 4 shows the results for the replacement test. The behaviours are similar with the other types of degradation.

Unsurprisingly, all the error rates increase with the score modification rate. The main observation that can be made is that this increase is roughly the same for every considered system. This shows that possible score errors deteriorate the performance, but have no significant influence on the choice of the optimal low level layer.

## 5. CONCLUSION

This study compares different instances of low-level models used in audio-to-score alignment through an extensive experimental evaluation where numerous variants of the models components (especially features and likelihood calculation) and their parameters are tested. In order to be robust to error-prone scores, our alignment systems infer the note durations without any other information than the order of the chords in the piece. Our experiments show that among the various features used in previous alignment proposals, some are significantly more efficient than others when correctly parameterized, and that the choice of the feature is all the more important when it is applied to polyphonic multi-instrumental music.

Whereas explicit spectral models are efficient in the monophonic case, their performance drops dramatically on the polyphonic data. It is proven that features using a logarithmic frequency scale are more effective than those based on an STFT.

The best chroma features perform at least as well as the semitone representations. Thus, CQT- or filterbank-based chroma vectors seem to be a fairly good choice of features, since the octave information does not appear to be essential. Furthermore, these representations are lighter than semitone features and more robust to possible octave errors in the score. The use of *synthesized templates* can also improve the system precision for most representations, as these templates are more realistic than the theoretical models.

In the continuation of this study, we will consider the use of other kinds of information, including onset or structure, and address the problem of the temporal model elaboration.

## 6. REFERENCES

[1] Arshia Cont, "A coupled duration-focused architecture for realtime music to score alignment," *IEEE Trans. on PAMI*, 2009, in press.

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

[3] M. Müller, F. Kurth, and T Röder, "Towards an efficient algorithm for automatic score-to-audio synchronization," in *Proc. of ISMIR*, 2004.

[4] C. Raphael, "Aligning music audio with symbolic scores using a hybrid graphical model," *Machine Learning Journal*, 2006.

[5] Se. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *ICASSP*, 2009.

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

[7] "Music information retrieval evaluation exchange 2006, score following task: `http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal`,".

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

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

[10] G. Peeters, "Musical key estimation of audio signal based on hidden markov modeling of chroma vectors," in *DAFx*, 2006.

[11] Y. Zhu and M.S. Kankanhalli, "Precise pitch profile feature extraction from musical audio for key detection," *IEEE Trans. on Multimedia*, vol. 8, no. 3, pp. 575–584, June 2006.

[12] E. Gómez, "Tonal description of polyphonic audio for music content processing," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 294–304, 2006.

[13] M. Goto, "Rwc music database: Popular, classical, and jazz music databases," in *Proc. of ISMIR*, 2002.