AUTOMATIC GENERATION OF LEAD SHEETS FROM POLYPHONIC MUSIC SIGNALS

Jan Weil, Thomas Sikora
Communication Systems Group
Technische Universität Berlin

J.-L. Durrieu, Gaël Richard
Institut Telecom
Telecom ParisTech
CNRS LTCI

ABSTRACT

A lead sheet is a type of music notation which summarizes the content of a song. The usual elements that are reproduced are the melody, chords, tempo, time signature, style and the lyrics, if any. In this paper we propose a system that aims at transcribing both the melody and the associated chords in a beat-synchronous framework. A beat tracker identifies the pulse positions and thus defines a beat grid on which the chord sequence and the melody notes are mapped. The harmonic changes are used to estimate the time signature and the down beats as well as the key of the piece. The different modules perform very well on each of the different tasks, and the lead sheets that were rendered show the potential of the approaches adopted in this paper.

1. INTRODUCTION

The lead sheet format is a convenient form of music notation for songs. It is mostly used for popular music and famously represented by collections of Jazz standards, e.g., *The Real Book*. It allows the musician to see all the important elements necessary to perform a song in a very compact format. It mostly consists of a single staff; the melody is notated in Western music standard, with the associated lyrics under the staff and the chord sequence noted above it. The lead sheet also often specifies the style, i.e., the way the melody has to be played, e.g., straight or swung rhythm, and the way the accompaniment should be generated from the chords. Of course, it also defines the time signature, the key and the tempo.

Very few works have been oriented towards producing usable music scores directly from audio. In [1], the authors estimate the melody, the bass line, and the chords. However, the results are not temporally quantized, so the output is not completely suited for lead sheet generation itself. This temporal quantization is indeed a non-trivial problem and we propose a potential solution in this paper.

The proposed lead sheet transcription system can be broken down into four seperate modules which exchange intermediate results. These modules are depicted in Fig. 1. The beat tracker provides a continuous pulse grid which forms the temporal basis for the other modules. The algorithm favours faster tempi such that the risk of phase errors is minimized and ensures a continuos beat grid. The reader is refered to [2] for details about the chosen approach. In this article, we directly use the output of this algorithm. The i^{th} beat position in seconds is denoted b_i. The harmonic module estimates beat-aligned chord sequences, the most likely measure grid, and the key of the piece. The measure grid is in turn used to refine the chord sequence by making chord change probabilities depend on the position in the measure. The chord detection module is based on the approach described in [3]. The melody module first separates the main melody and the accompaniment building on the approach presented in [4]. The model is extended such that the fundamental frequencies of the main melody and the musical (MIDI) notes of the melody are jointly estimated. The rendering module determines the appropriate time signature, quantizes the note onsets and durations of the melody to sub-divisions of the beat level, divides both melody and chords in measure blocks, and applies pitch spelling depending on the estimated key.

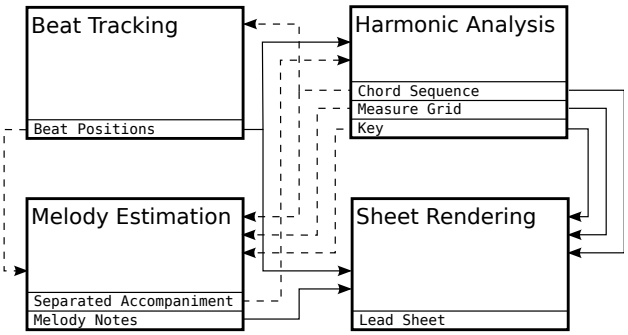

Figure 1. Modules of the proposed system along with the intermediate results they exchange. Dashed lines mark potential future dependencies.

In the following section we describe the chord detection scheme and how the down-beat positions are estimated using the detected chord sequence. After that the key estimation method is introduced. The melody extraction is discussed in Sec. 4. In Sec. 5, we describe how the lead sheets are rendered. Finally, we present the results as well as our conclusions and perspectives.

2. ESTIMATION OF CHORDS AND MEASURES

2.1 Chord detection

The chord detection module can be considered one of the numerous followers of the approaches described in [5] and [3], which are based on Hidden Markov Models (HMM). We model the chords as states of the HMM using a chord alphabet comprising major and minor chords, i.e., the ergodic model has $S = 24$ states ω_k, $k \in [1, S]$. The chord sequence is given as the most likely sequence of states given the observed feature sequence; this is known as the decoding problem which is solved using the Viterbi algorithm. Training and decoding is done in a 10-fold cross-validation setup.

2.1.1 Feature extraction

Beat-synchronous chroma vectors computed from the audio data form the observable features. The audio signal is mixed to a single channel and downsampled to 11025 Hz. We compute a constant-Q spectrogram [6] from note E2 (82.4 Hz) to note D#6 (1.24 kHz) using a hop size of 512 samples [1]. Due to the chosen lowest frequency the length of the longest window is 4096 samples. Chroma vectors are computed by summing up the magnitude of the transform for each of the 12 pitch classes over all four octaves. We then use the result from the beat tracking module to average all feature vectors within beat boundaries. Let $\boldsymbol{x}_c(i)$ denote the 12-dimensional chroma vector representing the time segment between beat positions b_i and b_{i+1}, $i = 1, 2, ...$

2.1.2 Training

The observation distribution is modeled as a multivariate Gaussian per state with mean vectors $\boldsymbol{\mu}_k$ and (full) covariance matrices $\boldsymbol{\Sigma}_k$, $k \in [1, S]$. The prior probabilities are considered uniformly distributed. Both the transition probabilities and the observation distribution are computed from the training sets using beat-quantized annotation data in a similar fashion as described for methods 1 and D in [7].

2.1.3 Initial chord sequence decoding

In the first stage, the chord sequence is decoded using the classic Viterbi algorithm. Let $q_1(i)$ denote the initial estimate of the decoded chord symbol which is assumed to have emitted $\boldsymbol{x}_c(i)$. Based on this initially decoded sequence we estimate the measure grid.

2.2 Estimating the measure grid

We assume that the probability of chord changes depends on the position in a measure and that, generally, chords are more likely to change at the beginning of measures [8]. We also assume a constant time signature; we do not, however, assume a 4/4 meter (although it clearly dominates our database). We consider a set of measure grid candidates of width $\nu \in [3, 4, ..., 8]$, i.e., each third, fourth, ..., eighth beat is assumed a down-beat. For each ν we have to consider ν potential phase candidates $\phi \in [1, \nu]$, i.e., the first down-beat is b_1, b_2, ..., or b_ν. For each of these grid width and phase candidate pairs, we compute the score

$$s(\nu, \phi) = T_{cc}(\nu, \phi) - F_{cc}(\nu, \phi), \tag{1}$$

where $T_{cc}(\nu, \phi)$ denotes the number of grid points which fall on beat positions with a chord change, i.e., $q_1(i-1) \neq q_1(i)$, and $F_{cc}(\nu, \phi)$ denotes the number of grid points without chord changes. The pair $(\nu_o, \phi_o) = \arg\max s(\nu, \phi)$ determines the chosen measure grid. Note that ν does not necessarily correspond to the numerator of the time signature as the beat we tracked may actually reflect half-time or double-time tempo.

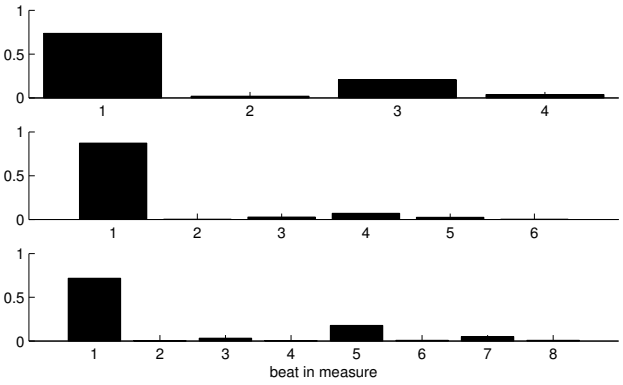

Figure 2. Probability of chord changes depending on the position in the measure for 4, 6, and 8 beats in a measure, respectively.

2.3 Refined chord sequence decoding

We use the measure grid estimate to compute the refined chord sequence $q_2(i)$ by making the transition probabilities depend on the position in the measure. Based on the down-beat information given by the annotation we compute the distribution of chord change positions relative to the measures from the training set. For the database we used, which will be discussed in Sec. 6, there are three possible values of ν: 4 (4/4 meter), 6 (6/8 meter), and 8 (4/4 meter; beat represents 8th notes). Fig. 2 depicts an example for the resulting probability profiles. As anticipated, chords are most likely to change on the beginning of a measure. We now propose a modified Viterbi decoding procedure. As we assume a continuous beat and measure grid, we can compute the current beat position in a measure $b_m = (i - \phi_o) \pmod{\nu_o} + 1$. Now the transition probability matrix is modified in the following manner: the diagonal, i.e., the probability to remain in the current state, is set to $1 - p_{cc}(b_m)$, where $p_{cc}(b_m)$ denotes the probability of a chord change at beat position b_m in the measure. The remaining non-diagonal elements are scaled such that they add up to $p_{cc}(b_m)$. Decoding the HMM using these varying transition probabilities gives the refined chord sequence $q_2(i)$.

[1] Note that, for the database we used, we can consider all pieces perfectly tuned to A4 = 440 Hz

3. KEY ESTIMATION

To estimate the key one can compute an average chroma profile and correlate it to key-specific templates [9]. Instead, we propose to compute the mean vector of the chord likelihoods using the trained Gaussian distributions for the chord states. We compare both approaches. We train key template profiles for major and minor keys which are circularly shifted to form all 24 possible key profiles. To this end, $\boldsymbol{x}_c(i)$ is circularly shifted such that the key is mapped to the root C for all pieces in the training set. The chroma-based templates $\boldsymbol{\mu}_{K1}(m)$ for both key modes m, major and minor, are computed as the mean vector of all shifted chroma vectors representing mode m. These templates have dimension 12. For the chord-based templates, the multi-variate Gaussian distribution is evaluated to compute the likelihoods $P(\boldsymbol{x}_c|\omega_k)$. The 24-dimensional mean vector of these chord likelihoods for both modes m, normalized to add up to one, gives the second set of key templates $\boldsymbol{\mu}_{K2}(m)$. To estimate the key of a piece we compute both the mean chroma vector and the normalized mean chord likelihoods. Then we compute the dot product of these test profiles and all 12 shifted variants of the two key templates as a measure of correlation. Note that for $\boldsymbol{\mu}_{K2}(m)$ the two halves of the likelihood vectors representing major and minor chords must be shifted independently. The key for which the template maximizes the dot product is chosen. This is done for both $\boldsymbol{\mu}_{K1}(m)$ and $\boldsymbol{\mu}_{K2}(m)$ to compare the results.

4. MAIN MELODY ESTIMATION

4.1 Global model for main melody sequence

Our model for melody estimation is based on the model proposed in [4]. In order to achieve a meaningful quantization of the desired melody line, we adapted the note duration model initially proposed in [10].

The observation audio signal x is considered as the instantaneous mixture of two contributions, the main instrument voice v playing the main melody and the accompaniment or background music m, i.e., $x = v + m$. This relation stays valid for the short time Fourier transform matrices X, V and M of these signals. We assume that the signal was decomposed into N frames, with Fourier transforms of F positive frequency bins. We model the complex Fourier transforms as complex proper centered Gaussians, for which we more specifically model the variances.

On one hand, for the accompaniment M, the "Nonnegative Matrix Factorization" (NMF) model is retained. The resulting variance $S_{M_n}(f)$ for $M_n(f)$, at frame n and frequency f is then given by:

$$S_{M_n}(f) = \sum_{r=1}^{R} W_M(f,r) H_M(r,n), \qquad (2)$$

where R is the number of elements in the spectrum dictionary W_M and H_M is the activation coefficient matrix associated to W_M. In matrix notation, with the variance matrix S_M such that $S_M(f,n) = S_{M_n}(f)$: $S_M = W_M H_M$.

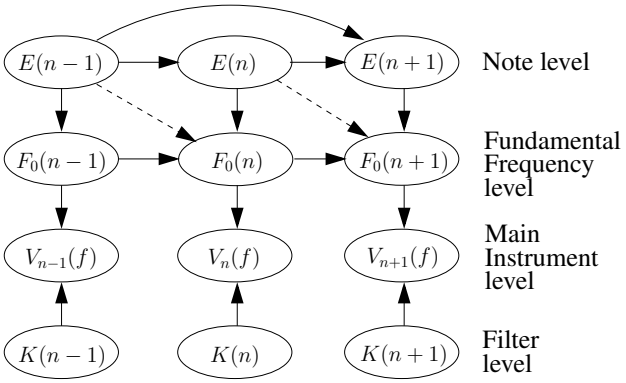

Figure 3. Generative model for the main instrument source/filter model.

On the other hand, the main instrument voice V is modelled through a source/filter model. The source part is driven by a three-layer generative model, shown on the upper part of Fig. 3. The filter part is modelled thanks to a two-layer model (lower part of Fig. 3). Note that the main instrument level V is also a hidden layer which, along with the accompaniment level M, gives the mixture observation level X.

The source level comprises two hidden levels. First, the fundamental frequency level $F_0(n)$ controls the pitch of the main instrument. These variables are dependent on the second layer, the note level. The evolution between the states of the note level $E(n)$ and the fundamental frequency states are explained in Sec. 4.2.

The filter layer is simpler, because here, we are more interested in the note and frequency levels. Therefore, we allow more flexibility in the evolution of the filter part and do not model any constraint on the corresponding sequence.

The main instrument level is then generated with the filter and fundamental frequency levels. The variance matrix S_V for $V_n(f)$, such that $S_V(f,n) = S_{V_n}(f)$, is given by:

$$S_V = \underbrace{(\overbrace{W_\Gamma H_\Gamma}^{W_\Phi} H_\Phi)}_{\text{Filter part}} . * \underbrace{(W_{F_0} H_{F_0})}_{\text{Source part}}, \qquad (3)$$

where W_Γ is a $F \times P$ dictionary of P smooth atomic elements, W_{F_0} a dictionary of N_{F_0} spectral combs for the voiced source part and H_Γ the coefficient matrix such that the actual filter dictionary $W_\Phi = W_\Gamma H_\Gamma$. The activation coefficient matrices for the filter and the source parts respectively are H_Φ and H_{F_0}.

The optimal note sequence $E = \{E(1), \ldots, E(N)\}$ is estimated within a Maximum Likelihood (ML) framework:

$$\hat{E}, \hat{F}_0, \hat{K} = \text{argmax}_{E,F_0,K} \log \text{p}(X, E, F_0, K). \qquad (4)$$

Such an estimation is computationally too intensive, and we propose in the next section some simplifications to estimate the different levels of the problem.

4.2 Model Approximations

In order to estimate the desired note sequence, we first neglect the constraint of having only one filter per frame. We then limit the problem to:

$$\hat{E}, \hat{F}_0 = \text{argmax}_{E,F_0} \log \text{p}(X, E, F_0), \quad (5)$$

The right-hand side of Eq. (5) can be further expressed as:

$$\log \text{p}(X, E, F_0) = \log \text{p}(X|F_0) + \log \text{p}(F_0|E) + \log \text{p}(E),$$

where, as shown on Fig. 3, we use that the sequence X is independent from E conditional on F_0. Furthermore, we assume that:

$$\log \text{p}(X|F_0) \approx \sum_n \log \tilde{H}_{F_0}(F_0(n), n). \quad (6)$$

In (6), the observation likelihood conditional on the melody fundamental frequency is approximated with a modified version $\tilde{H}_{F_0}$ of the source activation coefficient matrix H_{F_0} calculated on the data as described in [4]. During this first estimation round, the observation frames are assumed independent. We set $\tilde{H}_{F_0} = H_{F_0}$ and then normalize each column of $\tilde{H}_{F_0}$ by its maximum value.

The log-likelihood of the fundamental frequency sequence, conditional on the note state sequence, $\log \text{p}(F_0|E)$ is equal to:

$$\sum_{n=2}^{N} \log \text{p}(F_0(n)|F_0(n-1), E(n)) + \log \text{p}(F_0(1)|E(1))$$

Strictly speaking, $F_0(n)$ should also depend on $E(n-1)$, but for simplicity, we drop this dependency. We further assume that $\text{p}(F_0(n)|F_0(n-1), E(n))$ is proportional to the product:

$$\text{p}(F_0(n)|F_0(n-1)) \times \text{p}(F_0(n)|E(n)).$$

$\text{p}(F_0(n)|F_0(n-1))$ is a *prior* that simulates smooth f_0 variations. $\text{p}(F_0(n)|E(n))$ penalizes the distance between the fundamental frequency and the "expected" frequency for the note state $E(n)$. These functions are set to:

$$\text{p}(F_0(n) = f_2|F_0(n-1) = f_1) \propto \exp(-\alpha|\log_2(\frac{f_2}{f_1})|),$$
$$\text{p}(F_0(n) = f_0|E(n) = e) \propto \exp(-\beta|\log_2(f_0/f_e)|^2),$$

where f_e is the "standard" frequency for note $E = e$.
At last, we use the "segmental" duration model in [10] for the note state evolution:

$$\log \text{p}(E_{1:n}) = \log \text{p}(E_n|E_{1:n-1}) \log \text{p}(E_{1:n-1}). \quad (7)$$

The interested reader may find more information on this model in [10] , especially on the exact equations for the durations as well as on the beam searching algorithm that allows to find an optimal path for the sequence E.

To put it in a nutshell, we proceed as follows:

1. First assuming the independence of neighbouring frames, the parameters for the fundamental frequency and the filters are globally estimated.
2. We then extract pitch candidates for the main melody from the matrix H_{F_0} and use them to restrain the range of pitches to be tested when looking for the optimal path.
3. Finally, we find the optimal path of sequences E and F_0 using a beam search strategy, maximizing the approximated likelihood Eq. (5).

4.3 Generating a usable melody track transcription

The note sequence must be further quantized to produce a musical score. The fundamental frequencies are quantized onto the Western musical scale using the model for the sequence E. The temporal quantization is yielded to the rendering module such that the time signature can be considered.

5. LEAD SHEET GENERATION

Eventually, all the pieces of information are put together to render a readable transcription. Depending on ν_o and the estimated tempo we choose an appropriate time signature. Both the chords and the melody are processed in measure chunks. The onsets and the duration of the melody notes are quantized to a subdivision of quarter notes. These are usually eighth notes, which gives a good tradeoff between quantization errors and spurious notes. Depending on the estimated key, a simple pitch spelling algorithm is applied for both notes and chords. Basically, we choose note and chord names such that the distance on the circle of fifths is minimized.

6. RESULTS AND EVALUATION

In order to assess the different modules of the transcription system, we need a database for which the chords, the beat, and the melody line are annotated. Assembling such a database by manually annotating audio recordings is highly time-consuming. We found using the Band-In-A-Box [2] (BIAB) format a convenient way of generating the annotation in a semi-automatic way. BIAB is software which generates musical accompaniment given a sequence of chords, a tempo, and a style; it also supports melody tracks. Thus, BIAB files contain all the information which is relevant for the lead sheet generation task. Actually, BIAB even features lead sheet printouts, which gives a convenient reference for the subjective assessment of the results.

Our database comprises 278 files adding up to about 16.5 hours of audio material. It is a subset of the *Pop & Rock* database gathered by members of the Yahoo BIAB user group. Details are available on-line [11]. The files are rendered substituting the oboe for the singing voice, which is an instrument that shares a number of acoustic properties with the human voice. We used a modified version of one of the BIAB parsers available on-line to extract the relevant information from the BIAB files.

[2] http://www.band-in-a-box.com/

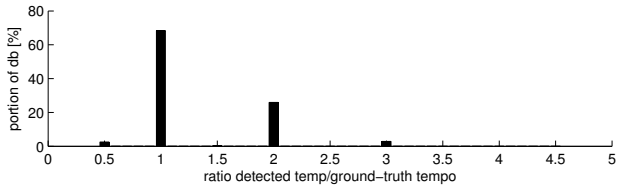

Figure 4. Histogram of the ratio *detected tempo / ground-truth tempo* over the entire database.

6.1 Beat tracking evaluation

We use the same metric as in [2] to evaluate the beat tracking module. The performance measure is the fraction of the longest continuous portion of the piece for which all beats are detected. A ground-truth beat is considered correctly tracked if the absolute distance to the nearest detected beat is smaller than 17.5 % of the period. If the ratio of the detected tempo to the ground-truth tempo is either two or three, we only consider every second or third beat, respectively, during the evaluation and choose the starting beat which maximizes the performance (see [2] for details). Fig. 4 depicts the histogram of the ratio *detected tempo / ground-truth tempo*. There is a single file for which the ratio is 1.5 which must be considered wrong. The average beat tracker performance is 94.1 %. For 91.4 % of all pieces we correctly track more than 90 % of the beats.

6.2 Down-beat tracking evaluation

The down-beat information implicitly given in the BIAB files cannot be trusted. Historically, BIAB's support for meters other than 4/4 is weak and sometimes the system is abused, e.g., a 6/8 meter would be recorded as a slower 4/4 meter where each beat of the 4/4 meter collects three beats of the 6/8 meter. Generally, the beat given in BIAB files is not guaranteed to correspond to the tactus period, i.e., the denominator of the time signature. It may reflect half-time tempo, double-time tempo, or ternary meters. To assess the proposed measure grid estimation approach we have to take these peculiarities into account. In compliance with the beat tracker performance measure we consider a down-beat correctly detected if the absolute distance to the closest ground-truth downbeat is less than 17.5 % of the period estimated by the beat tracking module. We compute the down-beat performance measure as the fraction of the longest continuous portion for which all down-beats were correctly detected. This is a particularly conservative measure as it combines both the result of the beat tracking module and the estimated measure grid based on detected chord change points. The average down-beat performance is 87.3 %.

6.3 Chord estimation evalution

We use basically the same evaluation measure as applied to the 2008 MIREX chord detection task [3] . All annotated chord symbols are mapped to their root triads resulting in five chord classes: major, minor, diminished, augmented, and suspended. (Note that 98.3 % of the chord symbols in our database fall into the major and minor categories.) This results in $5 \cdot 12 + 1$ possible states, including the no-chord state, which is used in the two pickup bars. The evaluation measure is the overlap in seconds between the detected chord sequence and the ground-truth sequence mapped to the 61 possible states as described above. The average overlap for the entire database is 76.4 % for the initial chord detection phase and 79.3 % for the refined estimation using transition probabilities depending on the position in the measure. The average overlap quantized to beats, which is more relevant to the transcription task, is 80.0 %; it is 82.7 % when the pickup bars are discarded.

[3] MIREX 2008 Evaluation Campaign, website: http://www.music-ir.org/mirex/2008/

6.4 Key estimation evaluation

For transcription purposes, a confusion of relative major and minor keys does not matter as the key signature remains the same. To evaluate the key estimation algorithms we thus compute the difference in the numbers of sharp or flat symbols, i.e., the smallest distance on the circle of fifths either clockwise (positive) or counterclockwise (negative). Fig. 5 shows the histogram of the key error measure for both key estimation approaches over the entire database. Both approaches correctly estimate the key signature for the majority of the pieces. However, the portion of the database for which the absolute key signature error is not greater than one is 80.2 % using the chroma profiles and 93.5 % using the mean chord likelihoods. The chroma-based approach is prone to confuse minor keys with their relative major keys (+3), e.g., A major instead of A minor, or with the key of the (major) dominant in the case of harmonic minor (+4), e.g., E major instead of A minor. Examining the statistics reveals that the variance remains significant for the chroma profiles. One could try to use a Gaussian classifier instead but, here, the method using the mean chord likelihoods works very well. In Pop and Rock music the chord range of the diatonic scale is often extended to include chords of keys which are close on the circle of fifths, e.g., a major chord on the minor 7th degree of a major scale (subdominant of the subdominant); this explains absolute key signature errors of one.

6.5 Melody tracking evaluation

For the melody estimation, we selected 11 songs that fit our definition of the main melody. For each song, the melody estimation algorithm returns the transcribed notes of the melody, with their MIDI note number, onset and offset times. A transcribed note is considered correct if there is a note in the reference with the same MIDI note number of which the onset time is close to the one of the transcribed note. The absolute difference between these onset times should be less than 150 ms. We compute precision, recall, and f-measure, and we provide the score obtained using the perceptually motivated measures in [12]. On our database, we obtain average recall, precision and f-measure of, respectively, 63 %, 68 % and 63 %. The average perceptive F-measure is 69 %. Fig. 6 shows the box and whiskers for

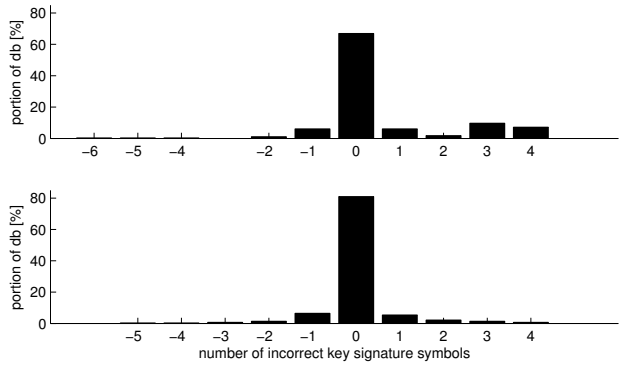

Figure 5. Histogram of the key signature error in steps on the circle of fifths for the chroma-based (top) and the chord likelihood-based method (bottom).

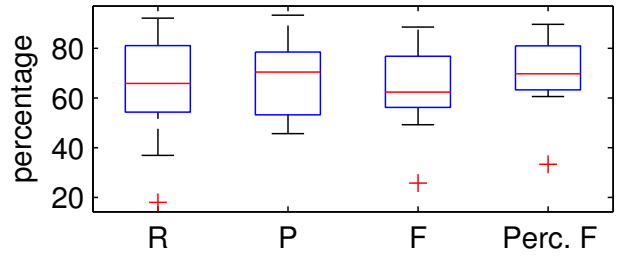

Figure 6. Box and whiskers plot of the results for melody estimation: Recall (R), Precision (P), F-measure (F) and perceptive F-measure (Perc. F).

the 11 songs. The outlier corresponds to a song for which the melody was too fast and too variable for the melody tracker to follow. The results are promising; however, the database we used was rather small and experiments on a bigger and more realistic database should be held in the future.

7. CONCLUSIONS AND PERSPECTIVES

We have proposed a lead sheet generation system. The tempo, time signature, chords, key, and melody were handled by several modules that can interact with each other. The chord sequence helps in determining the time signature, which in turn can be used to refine the chord sequence and also defines the minimum note duration for quantizing the melody. Our approach groups several modules that achieve state-of-the-art performance on each sub-task. Assessing the overall quality of the generated transcription is not trivial and subjective evaluation should be held for that purpose. For some examples available on-line [11] the resulting score is close to musician expectations. Some assumptions make the system targeted at Western music genres like Pop and Rock as represented by the chosen database. Evaluation of the sub-systems on real audio data remains to be done. The system could be further improved by allowing more joint estimations. A global model could cover all the aspects of the problem for which all the parameters for the different modules are jointly estimated. However, as for the melody module, such a model might be too complicated to be directly solved. Instead, this integration can be approximated for instance by including the detected beat positions in the melody note duration model. The melody estimation and separation can also be used to improve the chord sequence estimation.

8. ACKNOWLEDGEMENTS

This work was supported by the European Commission under contract FP6-027026 (KSpace), by the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement n° 216444 (PetaMedia) and by OSEO, French State agency for innovation, as part of the Quaero Programme.

9. REFERENCES

[1] M. P. Ryynanen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *CMJ*, 32(3):72–86, 2008.

[2] J. Weil, J.-L. Durrieu, G. Richard, and T. Sikora. Beat tracking using the delta-phase matrix. Technical report, Institut Telecom, Telecom ParisTech, CNRS LTCI, 2009.

[3] J.P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *ISMIR*, pages 304–311, 2005.

[4] J.-L. Durrieu, G. Richard, and B. David. An iterative approach to monaural musical mixture de-soloing. In *ICASSP*, pages 105–108, 2009.

[5] A. Sheh and D.P.W. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *ISMIR*, pages 185–191, 2003.

[6] J.C. Brown and M.S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *JASA*, 92:2698–2698, 1992.

[7] H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and hmm. In *CBMI'07*, pages 53–60, 2007.

[8] H. Papadopoulos and G. Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *ICASSP*, pages 121–124, 2008.

[9] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[10] E. Vincent. Musical source separation using time-frequency source priors. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 14(1):91–98, 2006.

[11] Accompanying website. http://www.nue.tu-berlin.de/research/leadsheets/.

[12] A. Daniel, V. Emiya, and B. David. Perceptually-based evaluation of the errors usually made when automatically transcribing music. In *ISMIR*, pages 550–555, 2008.