Integrating Additional Chord Information Into HMM-Based Lyrics-to-Audio Alignment

Matthias Mauch, Hiromasa Fujihara, and Masataka Goto

***Abstract*—Aligning lyrics to audio has a wide range of applications such as the automatic generation of karaoke scores, song-browsing by lyrics, and the generation of audio thumbnails. Existing methods are restricted to using only lyrics and match them to phoneme features extracted from the audio (usually mel-frequency cepstral coefficients). Our novel idea is to integrate the textual chord information provided in the paired chords-lyrics format known from song books and Internet sites into the inference procedure. We propose two novel methods that implement this idea: First, assuming that all chords of a song are known, we extend a hidden Markov model (HMM) framework by including chord changes in the Markov chain and an additional audio feature (chroma) in the emission vector; second, for the more realistic case in which some chord information is missing, we present a method that recovers the missing chord information by exploiting repetition in the song. We conducted experiments with five changing parameters and show that with accuracies of 87.5% and 76.7%, respectively, both methods perform better than the baseline with statistical significance. We introduce the new accompaniment interface *Song Prompter*, which uses the automatically aligned lyrics to guide musicians through a song. It demonstrates that the automatic alignment is accurate enough to be used in a musical performance.**

***Index Terms*—Audio user interfaces, hidden Markov models (HMMs), music, music information retrieval, speech processing.**

I. Introduction

LYRICS are the words to a song. In other forms of poetry or prose, too, order and rhythm are important to convey the meaning of the words, but only lyrics have the additional property of being synchronized with the music. If we consider a particular audio recording of a song, this alignment is the mapping that associates every word in the lyrics with the physical time at which it occurs in the recording. We call the task of producing the mapping *lyrics-to-audio alignment*. Human listeners—whether musically trained or not—can easily follow the given lyrics of a song, i.e., they mentally produce a lyrics-to-audio alignment; however, making this alignment explicit by annotating the physical time of every word is difficult and very time-consuming, which motivates the question whether it can be done automatically.

In the case of a solo (monophonic) vocal recording, lyrics-to-audio alignment is a special case of text-to-speech alignment (e.g., [27]), which is essentially solved. We consider here the more difficult case of polyphonic music: Regular popular music recordings in which the singing voice carrying the lyrics is but one of several instruments. The two problems in lyrics-to-audio alignment in polyphonic music that set it apart from the monophonic case are: To detect regions of vocal activity, and—in the parts where a singing voice is present—to recognize which part of the sound mixture corresponds to the singing voice.

Automatic lyrics-to-audio alignment has so far been solved only partially, as we explain in our review of existing methods below. Solutions to the problem have a wide range of commercial applications such as the computer-aided generation of annotations for karaoke or similar systems (e.g., *Song Prompter*, Section V), song-browsing by lyrics, and the generation of audio thumbnails [1], also known as audio summarization.

The first system addressing the polyphonic lyrics-to-audio alignment problem was a multimodal approach [26] (further developed in [11], [12], which deals with finding regions of vocal activity by preliminary chorus-detection and beat-tracking steps. However, the preliminary steps pose relatively strong assumptions on the form and meter (time signature) of the songs, thus limiting the scope of the method. A more general segmentation paradigm is the core of a paper that is concerned with segment-level lyrics-to-audio alignment [14]: First, an unconstrained structural segmentation is performed and the chorus section is determined by a clustering heuristic. Then, a vocal activity detection (VAD) step is used to decide which of the structural segments are vocal, and a dynamic programming algorithm is used to align the song parts as annotated in the lyrics to the song segments automatically extracted from audio. Note that this approach uses audio features only to determine the segmentation and the segment-wise vocal activity.

More audio-centric approaches aimed at word-level alignment employ a hidden Markov model (HMM) and forced alignment [2], [7], [22]: Chen *et al.* [2] use a VAD component to restrict alignment to vocal areas. Mesaros and Virtanen's HMM [22] use audio features based on the singing voice automatically segregated from the audio, but little attention is devoted to VAD: verse or chorus sections are manually selected. Both vocal activity detection and singing voice segregation are addressed in [7], where a left-to-right HMM architecture is used to align lyrics to audio, based on observed Mel frequency cepstral coefficients (MFCCs). We have chosen this more complete approach as a baseline method for our research (see Section III).

Manuscript received November 02, 2010; revised February 25, 2011; accepted May 23, 2011. Date of publication June 16, 2011; date of current version November 09, 2011. This work was supported by CrestMuse, CREST, and JST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Daniel Ellis.

The authors are with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305, Japan (e-mail: m.mauch@aist.go.jp; h.fujihara@aist.go.jp; m.goto@aist.go.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2159595

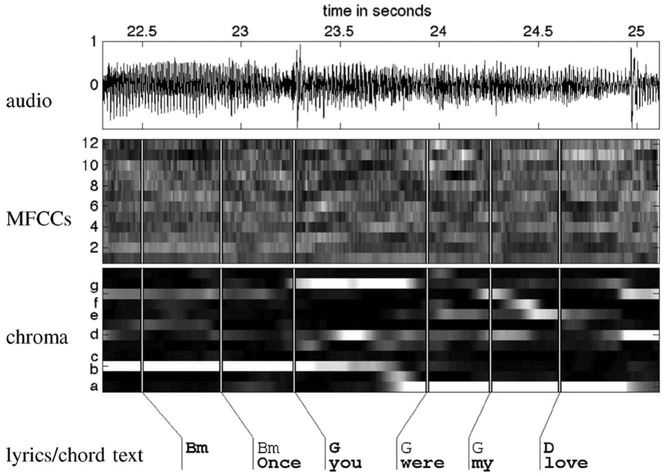
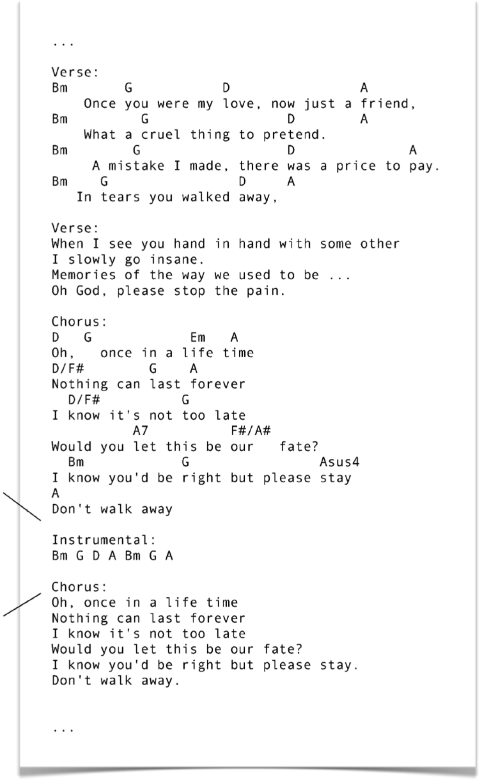


Fig. 1. Integrating chord information in the lyrics-to-audio alignment process (schematic illustration). The chords printed black represent chord changes, gray chords are continued from a prior chord change. Word-chord combinations are aligned with two audio features: an MFCC-based phoneme feature and chroma.


```
...

Verse:
Bm       G           D                A
    Once you were my love, now just a friend,
Bm         G                 D        A
    What a cruel thing to pretend.
Bm        G                  D              A
     A mistake I made, there was a price to pay.
Bm    G                D     A
   In tears you walked away,

Verse:
When I see you hand in hand with some other
I slowly go insane.
Memories of the way we used to be ...
Oh God, please stop the pain.

Chorus:
D   G            Em   A
Oh,   once in a life time
D/F#       G     A
Nothing can last forever
  D/F#          G
I know it's not too late
          A7          F#/A#
Would you let this be our   fate?
  Bm            G                Asus4
I know you'd be right but please stay
A
Don't walk away

Instrumental:
Bm G D A Bm G A

Chorus:
Oh, once in a life time
Nothing can last forever
I know it's not too late
Would you let this be our fate?
I know you'd be right but please stay.
Don't walk away.

...
```

first verse: all lyrics and chords given

subsequent verse: only lyrics; chords are omitted

blank line separates song segments

heading defines segment type

Fig. 2. Excerpt adapted from "Once In A Lifetime" (RWC-MDB-P-2001 No. 82 [10]) in the chords and lyrics format similar to that found in many transcriptions in song books or on the Internet.

Maddage *et al.* [15] turn the alignment task into an audio-to-audio synchronization task by using an audio representation of the lyrics instead of a feature model. This approach yields good word-alignment results (73.7%) under the assumption that the time stamps of all phrase beginnings are known in advance.

The existing lyrics-to-audio alignment systems have used only two information sources: the audio file and the lyrics. The main contribution of the present work is to integrate additional textual chord information into the lyrics-to-audio alignment framework as illustrated in Fig. 1. This information can be obtained from song books and the Internet websites such as "Ultimate Guitar"[1] in a format similar to the one given in Fig. 2.

Our goal is to show the following: additional chord information improves lyrics-to-audio alignment, and in particular the long-term alignment; chord alignment is not enough to provide satisfactory lyrics-to-audio alignment and is useful only in addition to phoneme alignment; partially missing chord annotations can be compensated for. We propose these two novel techniques:

1) an extension of a lyrics-to-audio alignment HMM which incorporates chords and chroma features, for the ideal case of complete chord information;
2) a two-step post-processing method that can recover missing chord information by locating phrase-level boundaries based on the partially given chords.

The rest of the paper is structured as follows. Section II describes the baseline HMM for lyrics-to-audio alignment without the use of chord information. Section III describes our novel extension of the HMM using chords and chroma, and also provides the results in the case of complete chord information. Section IV deals with the method that compensates for incomplete chord annotations by locating phrase-level boundaries, and discusses its results. In Section V, we present *Song Prompter*, an application based on our lyrics-to-audio alignment. Future work is discussed in Section VI, and Section VII concludes the paper.

II. Baseline Method

The baseline method [7] is based on an HMM in which each phoneme is represented by three hidden states, and the observed nodes correspond to the low-level feature, which we will call *phoneme feature*. Given a phoneme state, the 25 elements of the phoneme feature vector x_m consist of 12 MFCCs, 12 ΔMFCCs and 1 element containing the power difference (the subscript m stands for MFCC). For a phoneme state s, these $12 + 12 + 1$ elements are modeled as a 25-dimensional Gaussian mixture density $P_\mathrm{m}(x_\mathrm{m}|s)$ with 16 mixture components. The mixture models, and the transition probabilities between the three states of a phoneme are trained on Japanese singing (see Table I) using audio re-synthesized from estimated partial energies based on manually annotated fundamental frequencies of the main melody. For the use with English lyrics, phonemes are retrieved using the Carnegie Mellon University Pronouncing Dictionary[2] and then mapped to their Japanese counterpart. A left-to-right layout is used for the HMM, i.e., all words appear in exactly the order provided. The possibility of pauses between words is modeled by introducing optional "short pause" states, whose phoneme feature emissions are trained from the non-voiced parts of the songs.

Since the main lyrics are usually present only in the predominant voice, the audio is pre-processed to eliminate all other sounds. To achieve this the main melody voice is segregated in two steps: first, the predominant fundamental frequency is detected using PreFEst [9] (for a third-party evaluation, see [24]), along with the weights of its harmonics. Then, the harmonic

[1] http://www.ultimate-guitar.com and "Chordie" (http://www.chordie.com).

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

TABLE I
SIGNAL PROCESSING PARAMETERS OF THE TWO AUDIO FEATURES

	phoneme	chroma
Signal Processing		
sampling	16000 Hz	11025 Hz
frame length	25ms	372ms
window	Hamming	Hamming
frame rate	100 Hz	10 Hz (dupl.: 100 Hz)
Training Data [7]		
kind	Japanese pop (RWC) with phoneme annotations	n/a (expert model)
feature setting	re-syth'd from gr. truth F0	n/a (expert model)
number of songs	19	0 (no training)
result	$P(x_\mathrm{m}\|s)$	$P(x_\mathrm{c}\|s)$
Test Data		
kind	Anglophone pop (see Table II) with word onset annotations	
feature setting	several (see Section III-C)	
number of songs	20	

structure is used to re-synthesize the segregated melody line. The MFCCs necessary for the inference are extracted from the re-synthesized voice at intervals of 10 ms (details in Table I).

A second pre-processing step is the vocal activity detection (VAD) [6], which uses a simple probabilistic model with only two states (vocal or non-vocal) to find sung sections. The audio features used for this method are LPC-derived cepstral coefficients and ΔF0 (fundamental frequency difference).

The HMM is decoded using the Viterbi algorithm, during which the regions classified as non-vocal are constrained to emit only short pause states. This HMM is also a flexible framework which enables the integration of different features, as we explain below.

III. EXTENDING THE MODEL USING CHORDS AND CHROMA

This section presents our technique to align audio recordings with textual chord and lyrics transcriptions. We first provide motivation and technical details concerning the use of these transcriptions (Section III-A). In Section III-B we describe how the baseline HMM-based lyrics alignment system (Section II) is adapted for the additional chord information and the input of 12-dimensional chroma features. The results of the technique used in this section are given in Section III-C.

A. Textual Lyrics and Chords Annotations

Though there is no formal definition of the format used in the transcriptions appearing in song books and on the Internet, they will generally look similar to the one shown in Fig. 2. It contains the lyrics of the song with chord labels written in the line above the corresponding lyrics line. Chords are usually written exactly over the words they start on, and labels written over whitespace denote chords that start before the next word. In our example (Fig. 2) the lyrics of the verses are all accompanied by the same chord sequence, but the chord labels are only given for the first instance. This shortcut can be applied to any song segment type that has more than one instance, and transcribers usually use the shorter format to save space and effort. Song segment names can be indicated above the first line of the corresponding lyrics block. Song segments are separated by blank lines, and instrumental parts are given as a single line containing only the chord progression.

To show that the chord information does indeed aid lyrics alignment, we begin with the case in which complete chord information is given. More precisely, we make the following assumptions:

complete lyrics
Repeated lyrics are explicitly given;

segment names
The names of song segments (e.g., *verse*, *chorus*, ...) are given above every lyrics block;

complete chords
Chords for every song segment instance are given.

This last assumption is a departure from the format shown in Fig. 2, and in Section IV we will show that it can be relaxed.

B. HMM Network With Lyrics and Chords

After parsing the chords and lyrics file of a song, every word can be associated with a chord, the lyrics line it is in, and the song segment this line is part of. While only the word-chord association is needed for the HMM, the line and segment information retained can later be used to obtain the locations of lines and song segments.

The phoneme feature used in the baseline method bears little relationship with chord quality [25], and hence we have to use an additional audio feature: Chroma. Chroma is a low-level feature that relates to musical harmony and has been used in many chord and key detection tasks [8], [23] and for chord alignment [21], [25]. Chroma is also frequently used for score-to-audio alignment [3]. A chroma vector usually has twelve dimensions, containing activation values of the 12 pitch classes C, C#, ..., B. Our chroma extraction method [18] uses the original audio before melody segregation. It first calculates a pitch spectrum with three bins per semitone, which is then adjusted for minor deviations from the standard 440-Hz tuning. Then, the background spectrum (local mean) is subtracted and the remaining spectrum is further normalized by the running standard deviation, which is a form of spectral whitening. Finally, assuming tones with an exponential harmonics envelope, the non-negative least squares algorithm [13] is used to find the activation of every note, which is then mapped to the corresponding chroma bin. We use a 24-bin chroma representation in which the first 12 bins correspond to the bass region and the last 12 correspond to the melody region, which has been shown to increase chord identification [19]. Since chords change much more slowly than phonemes, the chroma method extracts features at a frame rate of 10 Hz (Table I), and to match the 100 Hz rate of the MFCCs we duplicate the chroma vectors accordingly.

The hidden states of the HMM are designed as in the baseline method, with the difference that every state now has two properties: the phoneme and the chord [Fig. 3(b)]. To model the state emissions we combine two different Gaussian feature models: the phoneme model $P(x_\mathrm{m}|s)$ as explained in Section III, and the chord model $P_\mathrm{c}(x_\mathrm{c}|s)$, which for state s models the bass and treble chroma using a 24-dimensional Gaussian density. The means of chord pitch classes are set to 1, all others to 0 and all variance parameters in the diagonal covariance matrices

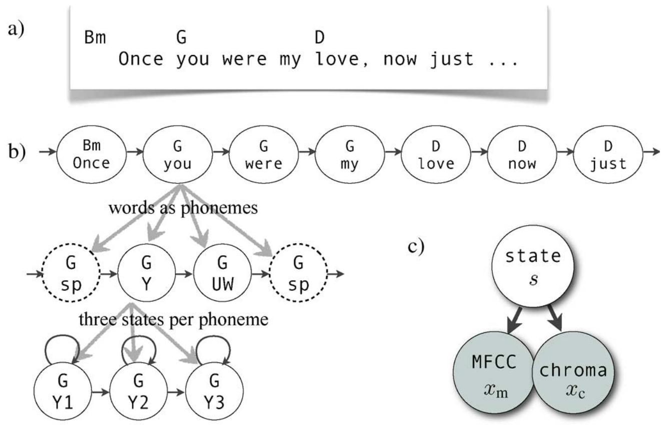


Fig. 3. HMM network with lyrics and chords. This example illustrates how the textual annotation (a) is turned into a left-to-right Markov chain (b), in which each chord/word HMM is decomposed into chord/phoneme HMMs, which in turn are decomposed into three states each. (c) illustrates that each such state has two emissions, one for the phoneme feature (MFCC) and one for chroma. Short pauses (sp) are optional, i.e., they can be skipped during Viterbi inference (dashed circles).

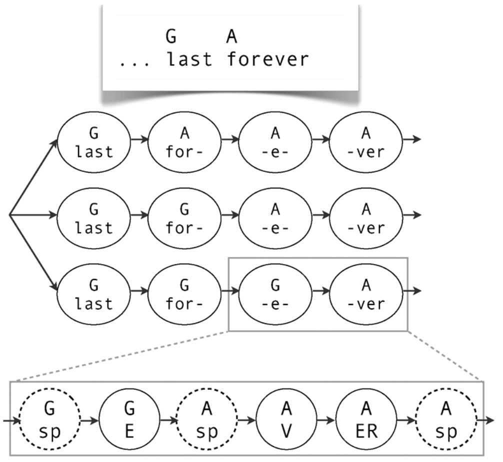


Fig. 4. Flexible chord onset (FCO). In this example, the word "forever" has three syllables and a chord change. FCO allows the chord to change at any syllable in the word, by taking one of the three alternative paths. The bottom of the figure shows how words are modeled as phonemes and short pauses. Short pauses are optional, i.e., they can be skipped during Viterbi inference (dashed circles).

are set to 0.2 [16]. The 121 unique chords are composed of 10 chord types (major, major with added 9th, major sixth, major seventh, dominant seventh, minor, minor seventh, diminished, augmented, suspended fourth) transposed to all 12 semitones, and one "no chord" type. They cover the large majority of chords found in popular music, and all other chords can be mapped to this set [16].

In order to unify the emissions streams into the final emission model $P(x|s)$ we use log-linear model combination as is customary in automatic speech recognition (e.g., [28]):

$$\log P(x|s) = a \log P_\mathrm{m}(x_\mathrm{m}|s) + b \log P_\mathrm{c}(x_\mathrm{c}|s). \quad (1)$$

The parameters a and b determine the relative weight of the two audio features, but also the audio features' weight in relation to the transition model. We test different combinations in Section III-C.

The textual representations of chord changes notated above the lyrics cannot be fully accurate, especially when the musical expression of the singer makes syllable boundaries ambiguous. Furthermore, even otherwise diligent transcribers often notate the chord at the beginning of the word, when the actual chord change occurs on a later syllable. We allow for this kind of variability in the annotation by designing a phoneme network as shown in Fig. 4, where chord changes in multi-syllable words are allowed to happen at any syllable boundary. We call this technique *flexible chord onset* (FCO).

TABLE II
SONGS USED FOR EVALUATION

	Artist	Song
1	ABBA	Knowing Me Knowing You
2	Bangles	Eternal Flame
3	Blondie	Call Me
4	Duffy	Warwick Avenue
5	Duran Duran	Ordinary World
6	Franz Ferdinand	Do You Want To
7	Shinya Iguchi (RWC)	Once In A Life Time
8	Shinya Iguchi (RWC)	Someday
9	Martika	Toy Soldiers
10	Muse	Guiding Light
11	Robert Palmer	Addicted To Love
12	Queen	We Are The Champions
13	Otis Redding	The Dock Of The Bay
14	Santana	Black Magic Woman
15	Simon and Garfunkel	Cecilia
16	Take That	Back For Good
17	Tina Turner	What's Love Got To Do With It
18	Toto	Africa
19	U2	With Or Without You
20	Zweieck	She

For inference we use an implementation of the Viterbi algorithm developed for the baseline method. The output of the Viterbi decoder assigns to every phoneme the estimated time interval within the song.

C. Results I

In order to evaluate the performance of our methods we chose the 20 anglophone songs listed in Table II (18 international pop songs and 2 songs[3] from the RWC Music Database [10]). We hand-labeled the physical onset time of every word in these songs. Previous work in lyrics-to-audio alignment has usually been evaluated only on phrase level, for which hand-labeling is less laborious, but the often uneven distribution of words over a lyric line makes the use of word-level timestamps a more meaningful ground truth representation.

Evaluation Metrics: Let N_songs be the number of songs in our test dataset, and N_k the number of words in the kth song. We evaluate the alignment according to the mean percentage

$$p_\tau = \frac{1}{N_\text{songs}} \sum_{\text{song } k} \underbrace{\frac{1}{N_k} \sum_{\text{word } i} \mathbf{1}_{|\hat{t}_i - t_i| < \tau} \times 100}_{= p_\tau^k - \text{ mean percentage over } k\text{th song}} \quad (2)$$

of start time estimates $\hat{t}_i$ that fall within τ seconds of the start time t_i of the corresponding ground truth word, averaged over

[3] RWC-MDB-P-2001 Nos. 82 and 84.

TABLE III
RESULTS I: ACCURACY AND MEAN ABSOLUTE DEVIATION FOR ALL 87 EXPERIMENTS. A FILLED CIRCLE DENOTES AN ACTIVATED FEATURE, THE "±" SIGN PRECEDES THE SAMPLE STANDARD DEVIATION VALUE (UNBIASED ESTIMATE)

				accuracy in %				mean abs. displacement in s			
phoneme				chroma weight b				chroma weight b			
weight a	PreFEst	VAD	FCO	0.0	0.5	1.0	1.5	0.0	0.5	1.0	1.5
0.0	○	○	○	–	57.2 ± 25.9	57.6 ± 26.2	57.6 ± 26.1	–	8.3	8.3	8.3
0.5	○	○	○	36.2 ± 21.6	78.6 ± 22.6	76.1 ± 27.4	73.1 ± 30.0	7.0	2.9	5.7	7.5
0.5	○	○	●	–	78.9 ± 22.6	76.5 ± 22.0	72.6 ± 25.4	–	2.9	3.3	5.8
0.5	○	●	○	40.3 ± 20.7	72.9 ± 22.5	70.7 ± 26.0	68.6 ± 28.6	5.3	3.3	6.0	6.3
0.5	○	●	●	–	69.8 ± 22.8	68.2 ± 22.5	67.1 ± 21.7	–	3.7	3.9	3.9
0.5	●	○	○	38.9 ± 21.7	86.1 ± 06.7	81.2 ± 19.7	80.9 ± 20.6	8.2	0.6	1.7	2.0
0.5	●	○	●	–	86.2 ± 06.2	80.4 ± 19.1	79.6 ± 18.8	–	0.6	1.8	1.8
0.5	●	●	○	44.6 ± 23.3	78.0 ± 15.1	75.4 ± 20.4	75.1 ± 21.1	6.5	1.6	2.4	2.7
0.5	●	●	●	–	76.3 ± 15.4	72.9 ± 20.5	72.2 ± 19.8	–	1.8	2.6	2.6
1.0	○	○	○	38.7 ± 21.1	85.9 ± 08.6	81.6 ± 16.7	79.1 ± 22.7	6.1	0.7	2.0	2.9
1.0	○	○	●	–	86.4 ± 07.6	82.8 ± 14.6	77.7 ± 22.3	–	0.7	1.9	3.1
1.0	○	●	○	43.2 ± 22.5	78.1 ± 15.2	76.1 ± 20.2	73.0 ± 22.7	4.6	1.5	2.2	3.4
1.0	○	●	●	–	75.5 ± 16.1	70.4 ± 22.9	69.4 ± 22.8	–	1.7	3.7	3.9
1.0	●	○	○	38.6 ± 22.0	87.4 ± 06.4	85.1 ± 10.3	84.6 ± 10.8	8.9	0.6	0.8	0.8
1.0	●	○	●	–	87.1 ± 06.6	84.5 ± 10.9	84.2 ± 10.7	–	0.6	0.9	0.9
1.0	●	●	○	46.3 ± 22.0	80.2 ± 11.9	78.5 ± 14.9	78.3 ± 14.8	4.9	1.5	1.6	1.6
1.0	●	●	●	–	78.8 ± 12.9	76.8 ± 15.3	74.1 ± 20.6	–	1.6	1.8	2.5
1.5	○	○	○	37.8 ± 22.9	86.0 ± 08.4	86.8 ± 08.2	82.0 ± 16.9	7.0	0.7	0.6	2.0
1.5	○	○	●	–	86.5 ± 07.9	86.3 ± 08.3	83.2 ± 14.9	–	0.8	0.7	1.9
1.5	○	●	○	43.7 ± 23.6	79.7 ± 11.8	78.1 ± 15.6	77.9 ± 15.5	4.9	1.1	1.6	1.6
1.5	○	●	●	–	78.6 ± 12.9	76.0 ± 16.3	73.7 ± 20.6	–	1.3	1.7	2.4
1.5	●	○	○	36.6 ± 21.8	85.7 ± 08.7	**87.5** ± 06.5	87.4 ± 06.2	9.3	0.8	0.6	0.6
1.5	●	○	●	–	84.1 ± 10.5	87.3 ± 06.4	84.7 ± 11.2	–	1.4	0.6	0.9
1.5	●	●	○	**46.4** ± 22.3	79.5 ± 12.3	80.1 ± 11.9	78.6 ± 14.9	4.9	1.5	1.5	1.6
1.5	●	●	●	–	78.2 ± 12.6	78.9 ± 12.7	76.5 ± 15.6	–	1.7	1.6	1.9

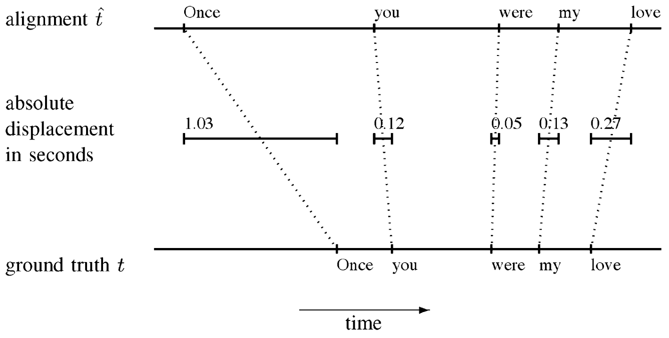


Fig. 5. Calculation of the performance metrics. In this example, the accuracy p_τ ($\tau = 1 \text{ second}$) from Equation (2) is 80% because four of the five words have an absolute displacement of $< \tau = 1 \text{ second}$. The mean absolute displacement d, see Equation (4), is 0.32 seconds, which is simply the arithmetic mean of the five absolute displacements.

songs. We use the p_τ, with $\tau = 1$ seconds, as a measure of the alignment performance and will simply call it *accuracy*. The unbiased estimate of the standard deviation of accuracy over songs is

$$s_\tau = \sqrt{\frac{1}{N_{\text{songs}-1}} \sum_{\text{song } k} \left(p_\tau^k - p_\tau\right)^2} \tag{3}$$

which provides a measure of the variation of accuracy between songs. The mean absolute deviation

$$d = \frac{1}{N_{\text{songs}}} \sum_{\text{song } k} \underbrace{\frac{1}{N_k} \sum_{\text{word } i} |\hat{t}_i - t_i|}_{\text{mean abs. displacement in } k\text{th song}} \tag{4}$$

between the time instant t_i at beginning of the ith target word and its estimate $\hat{t}_i$ (also averaged over all songs) allows an additional approach to the results. The metrics p_τ and d are illustrated in Fig. 5. All statements of significance will be made based on song-wise statistics.

Experimental Setup: We conducted experiments varying five different parameters: the phoneme feature weight a =0.0, 0.5, 1.0, 1.5, the chroma feature weight b =0.0, 0.5, 1.0, 1.5, the use of PreFEst melody extraction (on or off), the use of Vocal Activity Detection (VAD; on or off), and the use of flexible chord onset (FCO; on or off). Some of the 128 combinations effectively result in redundancies (e.g., PreFEst has no effect if the phoneme feature weight $a = 0.0$) and are hence omitted, leading to the set of 87 experiments whose results are displayed in Table III.

Significant Improvement Through Additional Chord Information: The overall best result in our tests was a accuracy of 87.5% (set bold in Table III-C), achieved with the use of additional chord information. The relatively low standard deviation of 6.5 percentage points indicates that the good results are stable over songs. More generally, Table III suggests that chord information improves accuracy substantially. To test whether this is true, we performed a Friedman test on the song-wise accuracy between the best-performing baseline method (accuracy: 46.4%, bold print in Table III; $a = 1.5$, $b = 0.0$, PreFEst on, VAD on, FCO off) and the worst-performing method that uses chord information in addition to the phoneme feature (accuracy: 67.1%; $a = 0.5$, $b = 1.5$, PreFEst off, VAD on, FCO on). The test confirms that the two methods perform significantly differently considering the accuracy $(p < 0.01)$. Comparing the baseline

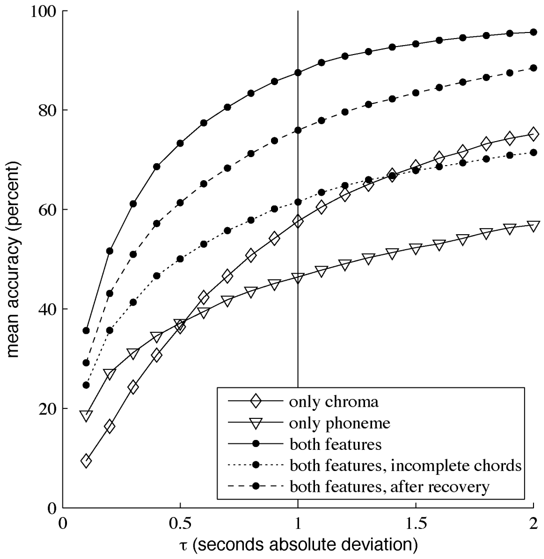


Fig. 6. Accuracies p_τ [see (2)] for different values of τ, from 0.1 seconds to 2.0 seconds. Three graphs with solid lines show the accuracies of the best-performing methods from the first experiment (Section III-C) for each audio feature combination, graphs with filled dots show accuracies for the three methods compared in the second experiment (Section IV-C). The gray vertical line marks the 1.0-s accuracy which we use to compare the results.

method to the method where only the chroma weight is changed to $b = 1.0$ (accuracy: 80.1%) reveals a larger difference, which is also significant according to a Friedman test ($p < 0.01$). Hence, we can be confident that the chord information improves performance. This is consistent with Fig. 6 which shows the accuracies p_τ for values of τ between 0.1 and 2.0 seconds for selected parameter setups.

Phoneme Feature Necessary for High Accuracy: As Table III furthermore suggests, the very good alignment scores of over 87% accuracy cannot be achieved with chord alignment alone, i.e., the phoneme feature is still necessary. This is indeed reflected in the significant increase in accuracy from 57.6% (best-performing method without phoneme feature; $a = 0, b = 1$) to 67.1% (see above) in the worst-performing method which uses chroma and phoneme features (Friedman $p = 0.025$).

The crossing lines in Fig. 6 show that the phoneme feature is indeed likely to be responsible for the fine alignment, while chroma provides the coarser large-scale alignment: for small accuracy thresholds $\tau < 0.5$ seconds using only the phoneme feature (triangle markers in Fig. 6) yields a higher accuracy. The method using only chroma (diamond markers) "overtakes" and becomes clearly better for values $\tau > 0.5$. The combination of both features (dot marker, solid line) leads to an even more striking difference. They work best together.

We would like to note that setting the parameter $a = 0.0$ reduces our method to plain chord alignment, where the phonemes influence in the HMM is limited to the duration distribution they imply. It may hence be argued that an ad-hoc heuristic that distributes phonemes or syllables within a chord span may lead to a satisfactory result. It is beyond the scope of this paper to implement such a heuristic, but it is unlikely that it could achieve competitively accurate results: we could not expect it to be significantly better than any of our methods that do use phoneme features, and these already have significant differences between each other (e.g., a =0.5, 1.5: Friedman p =0.01).

Influence of Parameters When Both Audio Features are Present: Since we have established the significance of both chroma and phoneme features for high accuracy, we now investigate the influence of the remaining three parameters when both phoneme and chroma features are present ($a > 0.0$ and $b > 0.0$). In order to show differences over these remaining methods we used a five-way analysis of variance (ANOVA) on the song-wise accuracy values (over the five parameters a, b, PreFEst, FCO, VAD). The rest of this paragraph discusses the results of this analysis. The use of PreFEst melody extraction leads to a significant increase in accuracy ($p < 0.01$), which was expected, since the detrimental influence of the accompaniment is reduced. This effect outweighs the effect of reduced salience of consonants in the re-synthesized melody. The effect of the flexible chord onsets is not significant ($p \approx 0.05$). We must conclude that the additional flexibility has compromised the benefits of this more faithful modeling of the joint features. The use of VAD leads to a significant decrease in accuracy ($p < 0.01$). This last outcome is surprising, since in the baseline a method using VAD achieves the highest accuracy (46.4%, set bold in Table III-C). The reason is that VAD allows the occurrence of a word only in regions identified as "vocal': VAD achieves an improvement when no coarse alignment via chords is available, even though some sections may be falsely classified as non-vocal. However, when additional chord information is used, it also provides this coarse alignment and VAD becomes obsolete; regions erroneously detected as non-vocal[4] will then decrease results.

In summary, the integration of chord information and chroma has largely improved performance, and is most effective when combined with the baseline phoneme feature. Among the other parameters tested, VAD is useful when used in the baseline method, and performing pre-processing melody extraction using PreFEst as a pre-processing step to the phoneme feature extraction significantly enhances accuracy.

IV. Recovering Partially Missing Chords

As we have seen in Fig. 2, among all verses (or choruses, etc.) it is usually only the first one that is annotated with chords. Our method presented above cannot be applied directly anymore because in the remaining segments it is no longer clear which chord to associate with which word. We will now consider this more difficult case by replacing the "complete chords" assumption given in Section III by a weaker assumption that is more in line with real world annotations.

Incomplete chords

Chords are given for the first occurrence of a song segment; subsequent occurrences of the same segment type have no chord information. They do still have the same number of lyric lines.

Transcriptions such as the one shown in Fig. 2 now comply with our new set of assumptions.

[4] The mean VAD recall averaged over songs is 0.817 (standard deviation: 0.093), which means that it misses around 18.3% of vocal regions. The mean precision is 0.877 (standard deviation: 0.154).

We take the model with the highest accuracy from Section III ($a = 1.5$, $b = 1.0$, PreFEst, no FCO, no VAD), which depends on chords and chroma, and apply it to the case of incomplete chords. We simply use the "no chord" model for words with missing chords, which ensures that no preference is given to any chord. As could be expected, the scarcity of information leads to a substantial performance decrease, from 87.5% (as discussed in the previous section) to 63.2%. Clearly, the partial chord information is not sufficient to maintain a good long-term alignment over the whole song. However, the first occurrence of a song segment type such as a verse is always given with lyrics and chord information, and we have shown in Section III-C that alignment performance is generally good when both features are used, so it would be likely to find good alignment at least in the song segments for which chord information is not omitted. This is indeed the case: if we restrict the evaluation to the song segments annotated with chords, we obtain a higher level of accuracy: 70.6%. (The accuracy is lower than in cases with full chord information because the alignment of chord progressions becomes ambiguous. This is especially likely when chord progressions are repeated: the alignment may "snap" to the wrong repetition. This happens, for example, in Muse's *Guiding Light*, song 10, where the bridge is aligned to the ending of the second verse because the chord progression is very similar.) This acceptable accuracy has motivated us to implement the following two steps.

1) Phrase-level segmentation: the results of the alignment are used to build a new chord-based HMM which models the chord progressions of phrases with known chords.
2) Constrained alignment: The phrase-level segmentation result is fed back to the original alignment HMM: Inference is performed constrained by phrase location.

Sections IV and IV-B will explain these steps in more detail and Section IV-C presents the results.

A. Phrase-Level Segmentation

In this first post-processing step we build a new HMM based entirely on chords and chroma, with three hierarchical levels as depicted in Fig. 7: Chord, song segment, and song. Based on the accuracy of 70.6% mentioned above, we assume that in segments with complete chord information the word time estimates are most often close to their true position. Since the words are associated with chords, they provide us with an estimate of the chord lengths for every segment type. Please note that even in cases where the alignment fails (i.e., it "snaps" to a different position in the song) the order of the chord progression is preserved. For each segment with complete chords we use these chord lengths to specify a new segment-specific HMM as a left-to-right chord sequence. Chords that cross a lyric line boundary, as the one from the first to the second lyric line in Fig. 2, are duplicated such that a line always starts with a chord. This is necessary because otherwise the model based only on chroma observations would be forced to find an erroneous phrase beginning.

The model of each chord is determined by its length ℓ in seconds: it is modeled by $\lceil 2\ell \rceil$ states, i.e., two states per second. Only self-transitions or transitions to the next state are allowed [see Fig. 7(a)]. The self-transition probability is $s = 0.2$. Hence, the expected duration of one state is 0.5 seconds at a frame rate of 10 Hz,[5] and the expected duration of the chord is $\lceil \ell \rceil$, i.e., the length estimated in the previous step, up to rounding. Of course, we could have modeled each chord with one state with a higher self transition probability, but that would have led to a geometrically distributed chord duration model and hence to a bias towards short durations. The chord duration in our implementation model follows a negative binomial distribution—similar to the one used in [17]—in which the probability of very short chord durations is reduced.

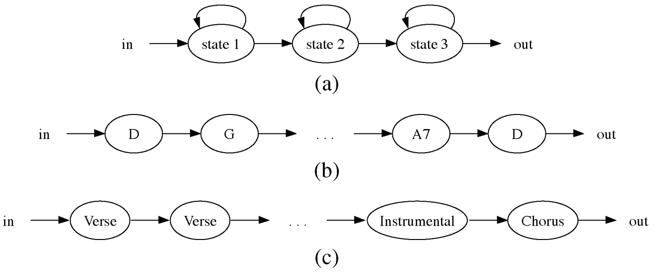


Fig. 7. HMM for phrase-level segmentation. Though the network is a strict left-to-right HMM, it can be thought of in terms of three hierarchical layers representing chords, song segments, and song structure. (a) Chord model: example of one chord of length $\ell = 1.5$ seconds. (b) Song segment model. (c) Song structure model.

The chord models are then concatenated to form a left-to-right model of the chord progression in one segment, as illustrated in Fig. 7(b). The segment HMMs are combined to the final left-to-right song HMM. Since we assume we know the names of all segments, and hence their succession, we can simply concatenate the individual segment HMMs in the correct order, as can be seen in the example in Fig. 7(c). Segment models may appear several times. Viterbi alignment returns estimates for chord change positions and—importantly—for phrase (lyric line) beginnings.

B. Constrained Alignment

This second post-processing step 2) combines the results obtained from the chord HMM in the previous step 1) with the initial HMM for lyric alignment. First, the HMM for lyrics and chord alignment is constructed in the usual way. In fact, we re-use the network and audio features from the initial alignment with incomplete chords. However, during inference we use the newly gained knowledge of line beginnings: we constrain the Viterbi search at frames of estimated line beginnings to the corresponding word in the lyrics, i.e., we "glue" the first words of a lyrics line to the estimated line beginning. This is equivalent to breaking up the song into phrases and aligning these separately. In some cases a line in the song starts with a chord but no simultaneous singing. The constrained inference does not prohibit this situation, since the word model always starts with an optional *short pause* (sp) state [Fig. 3(b)].

C. Results II

The model with the highest accuracy from Section III ($a = 1.5$, $b = 1.0$, PreFEst, no FCO, no VAD) is used for two further experiments with the same 20 songs (see Table II). First, the

[5] Since this HMM does not involve MFCCs we use the native chroma frame rate.

TABLE IV
ACCURACY FOR THE METHODS PRESENTED IN SECTION IV, AS EXPLAINED IN SECTION IV-C

method	accuracy in %
full chord information	87.5 ± 06.5
incomplete chords (no post-proc.)	63.2 ± 27.2
incomplete chords with recovery	76.7 ± 14.5

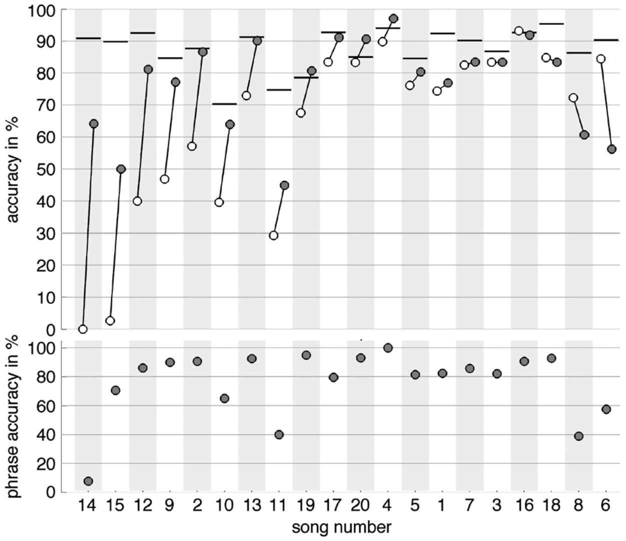


Fig. 8. Song-wise comparison. Upper graph: the lines connecting two points illustrate the improvement through the recovery method as explained in Section IV—blank circles represent accuracy with incomplete chord data (chords of repeated segments removed), and filled circles represent accuracy using our chord information recovery method. Black bars represent the accuracy obtained using all chord information (see Section III). Lower graph: phrase accuracy (word accuracy of first word in every line).

application of the original chord and lyrics alignment method without constrained alignment, and secondly the full method including the recovery steps 1) and 2). The accuracy results are given in Table IV, together with the result obtained under complete chord information. We observe that with respect to the method using complete chords, partially removing chord information clearly decreases accuracy [defined in (2)] from 87.5% to 63.2% (24.3 percentage points). Our proposed method, i.e., steps 1) and 2), can recover much of the lost information by applying phrase constraints, resulting in a accuracy of 76.7%, a significant increase (Friedman $p = 0.001$) of 13.5 percentage points.

Fig. 8 illustrates where the chord information recovery method presented in this section works best: the blank and filled circles connected by solid lines show the improvement from the method without post-processing (blank) to the method with chord information recovery (filled). The songs are sorted by amount of improvement, and we observe that the recovery method improves results for 15 of 20 songs. The differences are more substantial, if the accuracy of the method without recovery is low. For comparison, the bottom of the figure shows the phrase accuracy (calculated as accuracy over the first words in every lyric line). The mean phrase accuracy is 76.1% (standard deviation: 23.6%).

Some interesting details in Fig. 8 warrant a further qualitative explanation. The alignment of Santana's *Black Magic Woman*, and Simon and Garfunkel's *Cecilia* (songs 14 and 15) completely fails under partial chord information without recovery. In the case of *Cecilia*, for example, this is because the first annotated chorus part (with chords) "snaps" to the second instance of the chorus, thus corrupting the alignment in the all the surrounding parts. Incidentally, since the lyrics of the two choruses are identical, one could—under a more liberal evaluation metric—count the alignment on the second instance as correct. A second particular evident in Fig. 8 is that for Duffy's *Warwick Avenue* (song 4), U2's *With Or Without You* (song 19) and Zweieck's *She* (song 20) the recovery method performs better than the method with full chords. In all three cases the differences are very slight, i.e., the number of examples where the two alignments substantially differ is small. However, these examples suggest that in the method with full chord information erroneous phonemes locally dominated the alignment and "pulled" words into an adjacent line in some places. This was prohibited by the within-line alignment of the recovery method. As could be expected, an improvement requires good phrase-level alignment, and the three songs mentioned are indeed the ones with the best phrase accuracy (95.0%, 93.1%, and 100.0%, respectively) as shown in the bottom graph of Fig. 8. Conversely, the two only songs for which the recovery method leads to a substantial decrease in accuracy, *Do You Want To* by Franz Ferdinand and *Someday* by Shinya Iguchi (songs 6 and 8) show a low phrase accuracy (57.4% and 38.9%). Curiously, the song with the highest improvement, Santana's *Black Magic Woman* (song 14), also has a very low phrase accuracy (7.7%). This stems from the fact that every line in this song starts with a *pickup*, i.e., the first few sung notes precede the measure line and the chord change. The recovery method increases the likelihood of aligning the first word in the line to the beginning of the previous chord, and this is what happens in most cases: The line beginnings are estimated too early. However, the coarse alignment is still good, and the fine alignment "catches up," so that an overall accuracy of 64% is achieved.

As far as we know, this is the first time that a chord progression model of song segments has been applied for song segmentation, made possible by the partially given chord data. A particularly interesting feature is the capability of finding structures down to the phrase level, as the example in Fig. 9 demonstrates.

V. AN APPLICATION: USING ALIGNED CHORDS AND LYRICS IN *Song Prompter*

Through the use of additional chord information, our lyrics-to-audio alignment is now accurate enough to be useful in real world applications. In order to demonstrate this potential we developed the software system *Song Prompter* [20]. *Song Prompter* acts as a performance guide by showing horizontally scrolling lyrics, chords, beats marks, and bar marks in a graphical user interface, together with an audio accompaniment consisting of bass and MIDI drums. A song outline displays the song structure, including names and positions of sections for easy overview and navigation. *Song Prompter* enables users to sing and play live along the timeline of an original song, without having to memorize lyrics and chords or turning pages. Chord labels, and bass and audio playback can be transposed

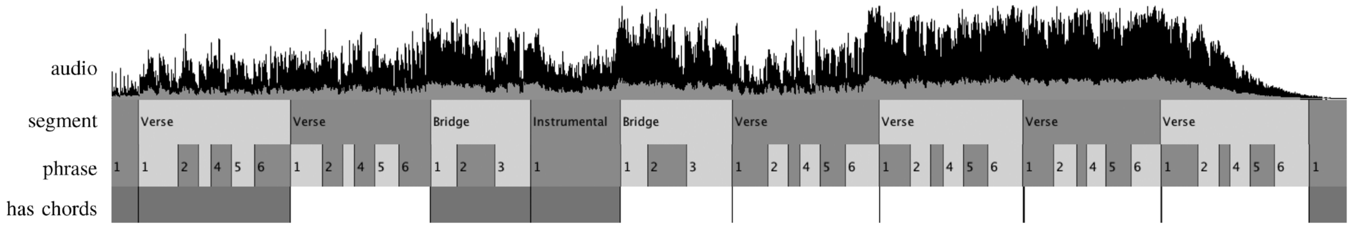
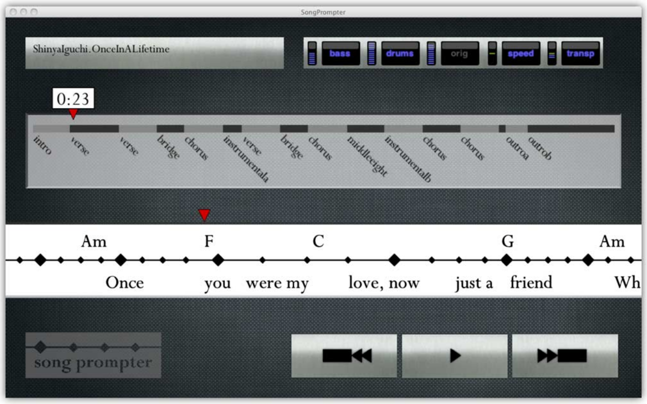


Fig. 9. Automatic segmentation as explained in Section IV. The top line is a representation of the audio waveform of the song *Eternal Flame* by the Bangles, with means and maxima of positive values indicated in gray and black, respectively. Below are the automatically detected segments, with names from the chords and lyrics annotation file. Underneath is the corresponding phrase-level segmentation (i.e., lyric lines). We can clearly see that the verse has six lines, the bridge has only three, while the instrumental section has no lyrics and hence no further segmentation. In the bottom line the segments for which chord information was available are shaded dark.



Am F C G Am
Once you were my love, now just a friend Wh

Fig. 10. *Song Prompter* screenshot: The gray Song Outline displays all parts of a song on a time line, providing easy access to any song part. The central performance guide of *Song Prompter* is the white Song Stream Pane, in which chords and lyrics are shown in the audio-aligned stream; the display is stretched to fit lyrics; the pulsating beat marks provide timing information and show the current song position; the red song position triangle can be dragged to the desired position. Chord symbols are transposed with the playback (here: −2 semitones).

to a different key, and the playback speed can be changed. A demonstration of *Song Prompter* can be viewed.[6]

The alignment method presented in Section III is the core of *Song Prompter*, and hence no musical score input is needed. This fact sets it apart from existing score following systems (for a review see [4]), karaoke systems, or musical computer games such as Rock Star and Guitar Hero.

A. Technical Realization

The display requires the physical onset times of chords, words, sections, beats, and bars in a song. Our method (Section III) is used to determine the physical onset times of chords, words, and song section in advance. The beat marks and larger bar marks complete the white song stream pane (see Fig. 10). Their positions are obtained using [5]. The fundamental frequency and the amplitude of the partials in the bass line are estimated using PreFEst [9].

At performance time the bass line is re-synthesized on the fly, using the frame-wise parameter estimates. Bass drum, snare drum and hi-hat MIDI notes are triggered, based on the extracted beat and bar times. The playback speed can be changed, and bass and original audio can be transposed.

B. Usage Scenarios

Many usage scenarios are conceivable. For example, in a cover band rehearsal situation, a band member can propose a new song by his favorite artist, and the band can immediately start playing the song based on the *Song Prompter* display of lyrics, beats and chord progression. When a band performs live, *Song Prompter* can literally act as an automatic prompter. In a more informal setting, it can be used when a party of friends want to sing together, and someone has brought along a guitar: *Song Prompter* is more convenient than song books because no pages need to be turned and everyone always knows the correct song position. Since the text is scrolling past, it can be displayed in a larger font than is possible in a static book format.

VI. Discussion and Future Work

Our chord information recovery method does not improve results for all songs, but in our experiments it did improve results for the majority of songs. No high-level music computing method can claim perfect accuracy, and systems that contain a number of successive steps suffer from errors that are propagated down to subsequent steps. We have presented such a system in this paper and are aware of this shortcoming. An approach that integrates both our proposed methods into one would be more elegant—and probably more effective. The main problem under partially missing chord data is that three song representations have to be aligned: lyrics, chords, and audio. Finding a model that encompasses all poses a significant challenge and we are not aware of standard statistical models that directly lend themselves to this task. Our flexible chord onset method shows that not all modifications to a more flexible model will immediately lead to improvements.

Further opportunities for future work arise from the front end of lyrics alignment methods. We have noted that melody extraction significantly improves alignment performance because the effect of the accompanying instruments is reduced. At the same time, however, consonants cannot be faithfully reproduced from the melody line and its harmonics. We assume, then, that combining phoneme features with and without pre-processing can combine the benefits of both: good vowel recognition through melody segregation/re-synthesis, and improved consonant recognition on the original waveform. A further possibility to extract the singing voice more clearly is by exploiting the

[6] http://www.youtube.com/user/SongPrompter.

stereo information in the signal, or, more generally, by source separation.

In the present study, the chord and lyrics files were checked and edited so they could be parsed unambiguously. For example, we made sure that the names of song segments were unambiguously recognizable as such so they would not be parsed as lyrics. In an application aimed at non-expert users, this "clean-up" would have to be performed automatically, i.e., the parsing of the files would have to be much more robust. Since the textual representation is generally aimed at human beings who already know the song, we expect that robust parsing requires the information contained in the audio, calling for an even broader integration-namely that of the alignment method with the parsing procedure.

Is it our next great goal, then, to develop methods for chord-aided alignment without any prior chord information and to automatically generate Internet-style chord and lyrics transcriptions? That would be a fantastic achievement, but it might be surpassed by the achievements of a class of new, multimodal music informatics systems which combine multiple audio features, information from the open Internet, local, and remote databases, and user interaction data. The methods presented here follow a similar approach. It remains to be seen whether future multimodal approaches can harvest the available sources of information in a meaningful way to perform music "understanding" tasks more reliably. Here lies one of the major challenges, but possibly the single greatest opportunity in music informatics over the next years.

VII. Conclusion

This paper has shown that additional chord information in a textual song book or Internet format can lead to substantially improved lyrics-to-chord alignment performance. This is true in the case in which chord information is provided for every part of the song, but also if the chords are only transcribed once for every song segment type (e.g., for the first of three verses), a shortcut often found in files in the Internet. We have proposed two methods that allow us to deal with these situations: the first one is based on an existing hidden Markov model that uses phoneme features for lyrics-to-audio alignment. We extend it by integrating chroma emissions and describe each hidden state in terms of the phoneme *and* the chord. We achieve an accuracy of 87.5% compared to 46.4% without chroma and 57.6% without phoneme features. Both differences are highly significant. Using melody extraction (PreFEst) as a pre-processing step for the phoneme feature extraction also significantly improves accuracy. If parts of the chord information are removed, the method performs worse (63.2%), though still better than the baseline method without chroma features. Our second proposed method succeeds in recovering much of the information lost: It uses the remaining partial chord information to build a new HMM with chord progression models for every song segment. Viterbi decoding of this HMM identifies the phrase structure of the song, so that lyrics alignment can be constrained to the correct phrase. This strategy significantly boosts accuracy by 13.5 percentage points to 76.7%.

We have shown that the performance of the phrase-level segmentation method is good (76.1%). This is the first time that segment-specific chord progression models have been used for segmentation and phrase-finding. Similar models may allow us to further relax assumptions on the chords and lyrics input format and hence to achieve robust performance in real-world situations.

We discuss the future directions of work, and more generally the challenges and opportunities created by new, multimodal approaches to music informatics that exploit the Internet, local and remote databases, and user interaction data in addition to audio features.

Acknowledgment

The authors would like to thank A. Klapuri, G. Richard, and D. Stowell for ideas and discussions. They would also like to thank the reviewers, whose suggestions have helped improve this paper.

References

[1] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 96–104, Feb. 2005.

[2] K. Chen, S. Gao, Y. Zhu, and Q. Sun, "Popular song and lyrics synchronization and its application to music information retrieval," in *Proc. SPIE*, 2006.

[3] R. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proc. Int. Comput. Music Conf. (ICMC 2003)*, 2003, pp. 27–34.

[4] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[5] M. E. P. Davies, M. D. Plumbley, and D. Eck, "Towards a musical beat emphasis function," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA'09)*, 2009, pp. 61–64.

[6] H. Fujihara and M. Goto, "Three techniques for improving automatic synchronization between music and lyrics: Fricative detection, filler model, and novel feature vectors for vocal activity detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'08)*, 2008, pp. 69–72.

[7] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic synchronization between lyrics and music CD recordings based on viterbi alignment of segregated vocal signals," in *Proc. 8th IEEE Int. Symp. Multimedia (ISM'06)*, 2006, pp. 257–264.

[8] T. Fujishima, "Real time chord recognition of musical sound: A system using common lisp music," in *Proc. Int. Comput. Music Conf. (ICMC 1999)*, 1999, pp. 464–467.

[9] M. Goto, "A real-time music scene description system: Predominant-F0estimation for detecting melody and bass lines in real-world audio signals," *Speech Commun. (ISCA J.)*, vol. 43, no. 4, pp. 311–329, 2004.

[10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical, and jazz music databases," in *Proc. 3rd Int. Conf. Music Inf. Retrieval (ISMIR 2002)*, 2002, pp. 287–288.

[11] D. Iskandar, Y. Wang, M. Y. Kan, and H. Li, "Syllabic level automatic synchronization of music signals and text lyrics," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, 2006, pp. 659–662, ACM.

[12] M.-Y. Kan, Y. Wang, D. Iskandar, T. L. Nwe, and A. Shenoy, "Lyric Ally: Automatic synchronization of textual lyrics to acoustic music signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 338–349, Feb. 2008.

[13] C. L. Lawson and R. J. Hanson*, Solving Least Squares Problems*. Englewood Cliffs: Prentice-Hall, 1974, ch. 23.

[14] K. Lee and M. Cremer, "Segmentation-based lyrics-audio alignment using dynamic programming," in *Proc. 9th Int. Conf. of Music Inf. Retrieval (ISMIR 2008)*, 2008, pp. 395–400.

[15] N. C. Maddage, K. C. Sim, and H. Li, "Word level automatic alignment of music and lyrics using vocal synthesis," *ACM Trans. Multimedia Comput., Commun., Applicat. (TOMCCAP)*, vol. 6, no. 3, 2010 [Online]. Available: http://doi.acm.org/10.1145/1823746.1823753, Article 19.

[16] M. Mauch, “Automatic chord transcription from audio using computational models of musical context,” Ph.D. dissertation, Queen Mary Univ. of London, London, U.K., 2010.

[17] M. Mauch and S. Dixon, “A discrete mixture model for chord labelling,” in *Proc. 9th Int. Conf. Music Inf. Retrieval (ISMIR 2008)*, 2008, pp. 45–50.

[18] M. Mauch and S. Dixon, “Approximate note transcription for the improved identification of difficult chords,” in *Proc. 11th Int. Soc. Music Inf. Retrieval Conf. (ISMIR 2010)*, 2010, pp. 135–140.

[19] M. Mauch and S. Dixon, “Simultaneous estimation of chords and musical context from audio,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1280–1289, Aug. 2010.

[20] M. Mauch, H. Fujihara, and M. Goto, “Song Prompter: An accompaniment system based on the automatic alignment of lyrics and chords to audio,” in *Proc. Late-Breaking Session 10th Int. Conf. Music Inf. Retrieval (ISMIR 2010)*, 2010, pp. 9–16.

[21] M. McVicar and T. De Bie, “Enhancing chord recognition accuracy using web resources,” in *Proc. 3rd Int. Workshop Mach. Learn. Music (MML10)*, 2010.

[22] A. Mesaros and T. Virtanen, “Automatic recognition of lyrics in singing,” *EURASIP J. Audio, Speech, Music Process.*, vol. 2010, 2010 [Online]. Available: , Article ID 546047.

[23] L. Oudre, Y. Grenier, and C. Févotte, “Template-based chord recognition: Influence of the chord types,” in *Proc. 10th Int. Soc. for Music Inf. Retrieval Conf. (ISMIR 2009)*, 2009, pp. 153–158.

[24] G. Poliner, D. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, “Melody transcription from music audio: Approaches and evaluation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[25] A. Sheh and D. Ellis, “Chord segmentation and recognition using EM-trained hidden Markov models,” in *Proc. 4th Int. Conf. Music Inf. Retrieval (ISMIR 2003)*, 2003.

[26] Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin, “Lyric Ally: Automatic synchronization of acoustic musical signals and textual lyrics,” in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 212–219.

[27] C. W. Wightman and D. T. Talkin, “The aligner: Text-to-speech alignment using Markov models,” in *Progress in Speech Synthesis*, J. P. H. Van Santen, Ed. New York: Springer, ch. 25, pp. 313–323, Progress in speech synthesis.

[28] A. Zolnay, R. Schlüter, and H. Ney, “Acoustic feature combination for robust speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2005, pp. 457–460.

Matthias Mauch received the *Diplom* degree in mathematics from the University of Rostock, Rostock, Germany, in 2005, and the Ph.D. degree in electronic engineering from Queen Mary, University of London, London, U.K., in 2010, for his work on the automatic transcription of musical chords from audio.

He is a Post-Doctoral Research Scientist at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests span a wide range of subjects related to music information retrieval, with a strong emphasis on music signal processing and the automatic detection of musical content in audio: chord transcription, key detection, beat-tracking, detection of the metric structure, harpsichord temperament classification, detection of vocal and instrumental solos, and the quantification and visualization of musical complexity.

Hiromasa Fujihara received the Ph.D. degree from Kyoto University, Kyoto, Japan, in 2010 for his work on computational understanding of singing voices.

He is currently a Research Scientist of the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include singing information processing and music information retrieval.

Dr. Fujihara was awarded the Yamashita Memorial Research Award from the Information Processing Society of Japan (IPSJ).

Masataka Goto received the Doctor of Engineering degree from Waseda University, Tokyo, Japan, in 1998.

He is currently the leader of the Media Interaction Group at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He serves concurrently as a Visiting Professor at the Institute of Statistical Mathematics, an Associate Professor (Cooperative Graduate School Program) in the Graduate School of Systems and Information Engineering, University of Tsukuba, and a Project Manager of the MITOH Program (the Exploratory IT Human Resources Project) Youth division by the Information Technology Promotion Agency (IPA).

Dr. Goto received 25 awards over the last 19 years, including the Young Scientists’ Prize, the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology, the Excellence Award in Fundamental Science of the DoCoMo Mobile Science Awards, the Best Paper Award of the Information Processing Society of Japan (IPSJ), and the IPSJ Nagao Special Researcher Award.