LYRICS-TO-AUDIO ALIGNMENT AND PHRASE-LEVEL SEGMENTATION USING INCOMPLETE INTERNET-STYLE CHORD ANNOTATIONS

Matthias Mauch **Hiromasa Fujihara** **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{m.mauch, h.fujihara, m.goto}@aist.go.jp

ABSTRACT

We propose two novel lyrics-to-audio alignment methods which make use of additional chord information. In the first method we extend an existing hidden Markov model (HMM) for lyrics alignment [1] by adding a chord model based on the *chroma* features often used in automatic audio chord detection. However, the textual transcriptions found on the Internet usually provide chords only for the first among all verses (or choruses, etc.). The second method we propose is therefore designed to work on these incomplete transcriptions by finding a phrase-level segmentation of the song using the partial chord information available. This segmentation is then used to constrain the lyrics alignment. Both methods are tested against hand-labelled ground truth annotations of word beginnings. We use our first method to show that chords and lyrics complement each other, boosting accuracy from 59.1% (only chroma feature) and 46.0% (only phoneme feature) to 88.0% (0.51 seconds mean absolute displacement). Alignment performance decreases with incomplete chord annotations, but we show that our second method compensates for this information loss and achieves an accuracy of 72.7%.

1. INTRODUCTION

Few things can rival the importance of lyrics to the character and success of popular songs. Words and music come together to tell a story or to evoke a particular mood. Even musically untrained listeners can relate to situations and feelings described in the lyrics, and as a result very few hit songs are entirely instrumental [2]. Provided with the recording of a song and the corresponding lyrics transcript, a human listener can easily find out which position in the recording corresponds to a certain word. We call detecting these relationships by means of a computer program *lyrics-to-audio alignment*, a music computing task that has so far been solved only partially. Solutions to the problem have a wide range of commercial applications such as the computer-aided generation of annotations for karaoke, song-browsing by lyrics, and the generation of audio thumbnails [3], also known as audio summarization.

The first system addressing the lyrics-to-audio alignment problem was a multimodal approach proposed by Wang *et al.* [4], which has since been developed further [5]. The method makes relatively strong assumptions on the form and meter (time signature) of the songs, which enables preliminary chorus-detection and beat-tracking steps to aid the final low-level lyrics alignment step. In [1] a left-to-right hidden Markov model (HMM) architecture is used to align lyrics to audio, based on observed mel frequency cepstral coefficients (MFCCs). Here too, several preprocessing steps such as singing melody segregation and vocal activity detection are employed, but these make fewer assumptions, and the more complex HMM models the evolution of single phonemes in time. Similar HMM-based approaches have been used in [6] and [7]. A special case of lyrics alignment based on the speech melody of Cantonese has been presented in [8]. These existing lyrics-to-audio alignment systems have used only two information sources: the audio file and the lyrics.

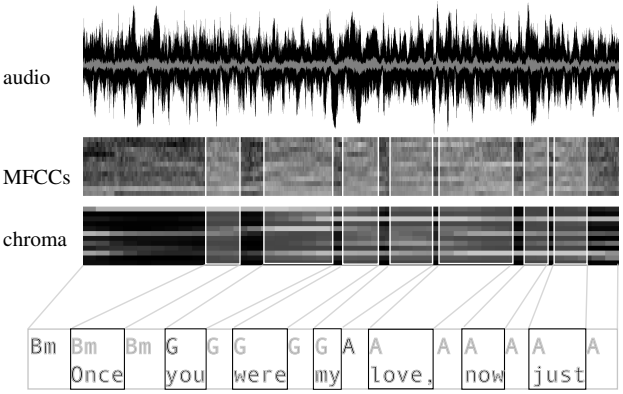


Figure 1: Integrating chord information in the lyrics-to-audio alignment process (schematic illustration). The chords printed black represent chord changes, grey chords are continued from a prior chord change. Word-chord combinations are aligned with two audio features: an MFCC-based phoneme feature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

In this paper we propose two novel techniques that integrate additional textual chord information (Figure 1) into the alignment framework:

1. an extension of the lyrics-to-audio alignment paradigm to incorporate chords and chroma features in the ideal case of complete chord information, and

2. a three-step method that can recover missing chord information by locating phrase-level boundaries based on the partially given chords.

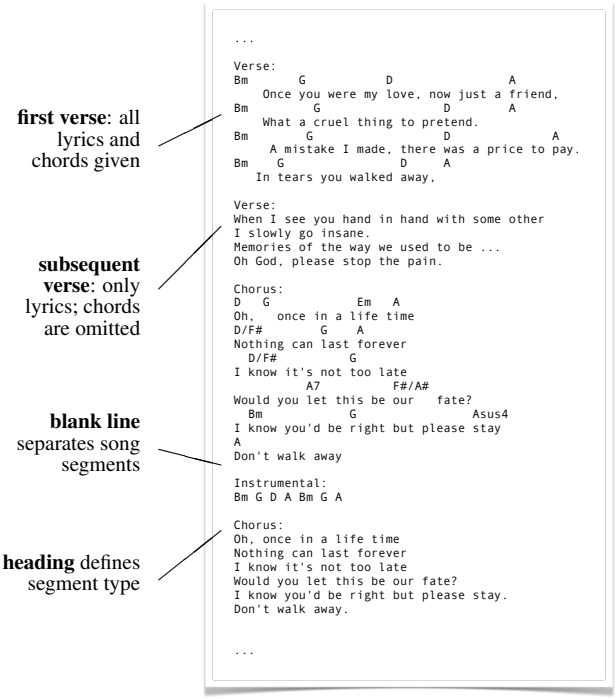


Figure 2: Excerpt adapted from "Once In A Lifetime" (RWC-MDB-P-2001 No. 82 [9]) in the chords and lyrics format similar to that found in many transcriptions on the Internet.

The motivation for the integration of chords is the vast availability of paired textual chord and lyrics transcriptions on the Internet through websites such as "Ultimate Guitar" [1] and "Chordie" [2] . Though there is no formal definition of the format used in the transcriptions appearing on the Internet, they will generally look similar to the one shown in Figure 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 (Figure 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.

The rest of the paper is structured as follows. Section 2 describes the hidden Markov model we use for lyrics alignment and also provides the results in the case of complete chord information. Section 3 deals with the method that compensates for incomplete chord annotations by locating phrase-level boundaries, and discusses its results. Future work is discussed in Section 4, and Section 5 concludes the paper.

[1] http://www.ultimate-guitar.com

[2] http://www.chordie.com

2. USING CHORDS TO AID LYRICS-TO-AUDIO ALIGNMENT

This section presents our technique to align audio recordings with textual chord and lyrics transcriptions such as the ones described in Section 1. To show that the chord information does indeed aid lyrics alignment, we start 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 Figure 2, and in Section 3 we will show that it can be relaxed.

An existing HMM-based lyrics alignment system is used as a baseline and then adapted for the additional input of 12-dimensional chroma features using an existing chord model [10]. We will give a short outline of the baseline method (Section 2.1), and then explain the extension to chroma and chords in Section 2.2. The results of the technique used in this section are given in Section 2.3.

2.1 Baseline Method

The baseline method [1] is based on a hidden Markov model (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*. To be precise, given a phoneme state s, the 25 elements of the phoneme feature vector x_{m} with the distribution $P_{\mathrm{m}}(x_{\mathrm{m}}|s)$ consist of 12 MFCCs, 12 ΔMFCCs and 1 element containing the power difference (the subscript m stands for MFCC). These 12+12+1 elements are modelled as a 25-dimensional Gaussian mixture model with 16 mixture components. The transition probabilities between the three states of a phoneme and the Gaussian mixtures are trained on Japanese singing. For use with English lyrics, phonemes are retrieved using the Carnegie Mellon University Pronouncing Dictionary [3] 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 modelled 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 three steps: first, the predominant fundamental frequency is detected using PreFEst [11]. The

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

	phoneme	chroma
sampling	16000 Hz	11025 Hz
frame length	25ms	372ms
window	Hamming	Hamming
frame rate	100 Hz	10 Hz

Table 1: Signal processing parameters of the two audio features.

estimated frequency at every 10ms frame is used to find the further harmonics, and the weights of the harmonics are computed. Finally, the resulting harmonic 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 10ms (details in Table 1).

A second pre-processing step is the vocal activity detection (VAD), 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 method is parameterized such that few vocal regions are missed, even if this causes some instrumental regions to be misclassified as vocal.

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.

2.2 HMM Network with Lyrics and Chords

In order to integrate chords in the baseline method described above we need to parse the chords and lyrics files, calculate a low-level harmonic feature (chromagram) and extend the HMM so that it can process the additional information.

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. In the present implementation a chord change on a word is assumed to start at the beginning of a word. 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.

Chroma is a low-level feature that relates to musical harmony and has been used in many chord and key detection tasks [12, 13], but only rarely for chord alignment [14]. Chroma is also frequently used for score-to-audio alignment [15]. A chroma vector usually has twelve dimensions, containing activation values of the twelve pitch classes C, C#,. . ., B. Our chroma extraction method [16] 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 [17] is used to find the activation of every note, which is then mapped to the corresponding chroma bin. Since chords change much more slowly than phonemes, the chroma method extracts features at a frame rate of 10Hz (Table 1), and to match the 100Hz 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. The observed emissions of the joint phoneme and chroma feature $x = (x_{\mathrm{m}}, x_{\mathrm{m}})$ are modelled by a probability distribution with log-density

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

where $P_m(x_m|s)$ is the baseline phoneme model, and $a = b = 1.0$ are weight parameters, which can be modified for testing. The subscripts m and c stand for MFCCs and chroma, respectively. $P_{\mathrm{c}}(x_{\mathrm{c}}|s)$ is the chord model, a set of 145 12-dimensional Gaussians that models the emissions of 145 different chords: 12 chord types (major, minor, major 7$^{\mathrm{th}}$,...) transposed to all 12 semitones, and one "no chord" type. The means of chord pitch classes are set to 1, all others to 0. All variance parameters in the diagonal covariance matrices are set to 0.2 [18].

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.

2.3 Results I

We ran two sets of eight experiments, one with vocal activity detection, and one without. In each set we varied the phoneme and chroma feature weights a and b in (1) within $\{0.0,\ 0.5,\ 1.0\}$ (the case in which both features have zero weight is omitted). In order to evaluate the performance of our method we chose 15 songs (13 pop hits and 2 songs [4] from the RWC Music Database [9]) with English lyrics (see Table 2) and hand-labelled every word in these songs with its onset time in seconds. Previous work in lyrics-to-audio alignment has been evaluated only on phrase level, for which hand-labelling is less laborious, but the often uneven distribution of words over a lyric line makes word-level alignment a more meaningful ground truth representation. We evaluate the alignment according to two criteria. The mean percentage

$$p = \frac{1}{N_{\text{songs}}} \sum_{\text{song } k} \underbrace{\frac{1}{N_{\text{words}}} \sum_{\text{word } i} \mathbf{1}_{|\hat{t}_i - t_i| < 1} \times 100}_{\text{average percentage over } k^{\text{th}} \text{ song}} \qquad (2)$$

of start time estimates $\hat{t}_i$ that fall within one second of the start time t_i of the corresponding ground truth word, averaged over songs. We will simply call this measure *accu-*

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

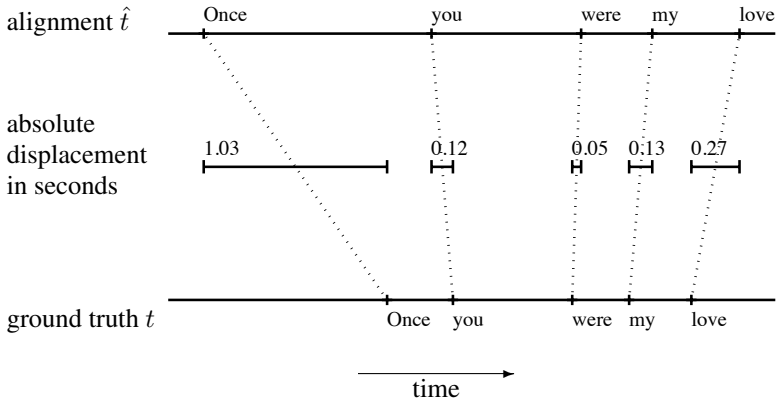


Figure 3: Calculation of the performance metrics. In this example, the accuracy p from Equation (2) is 80% because four of the five words have an absolute displacement of < 1 second. The mean absolute displacement d, see Equation (3), is 0.32 seconds, which is simply the arithmetic mean of the five absolute displacements.

	Artist	Song
1	Bangles	Eternal Flame
2	U2	With Or Without You
3	Robert Palmer	Addicted To Love
4	Martika	Toy Soldiers
5	Queen	We Are The Champions
6	Simon and Garfunkel	Cecilia
7	Otis Redding	The Dock Of The Bay
8	Shinya Iguchi (RWC)	Once In A Life Time
9	ABBA	Knowing Me Knowing You
10	Duran Duran	Ordinary World
11	Toto	Africa
12	Santana	Black Magic Woman
13	Shinya Iguchi (RWC)	Someday
14	Franz Ferdinand	Do You Want To
15	Duffy	Warwick Avenue

Table 2: The songs used for evaluation.

racy. A second measure is the mean absolute displacement

$$d = \frac{1}{N_{\text{songs}}} \sum_{\text{song } k} \underbrace{\frac{1}{N_{\text{words}}} \sum_{\text{word } i} |\hat{t}_i - t_i|}_{\text{mean abs. displacement in } k^{\text{th}} \text{ song}} \qquad (3)$$

between the time instant t_i at beginning of the i^{th} target word and its estimate $\hat{t}_i$, which is also averaged over all songs. Both metrics are illustrated in Figure 3.

Table 3 shows accuracy values, i.e. the mean percentage p as defined in Equation (2), for all tests. They can be summarized as follows: the highest accuracy of 88.0% is achieved with chroma and phoneme feature, but no vocal activity detection (VAD); the chroma features provide large scale alignment while the phonemes provide short-term alignment; VAD does improve results for the baseline method, i.e. when chroma is switched off. The next paragraphs provide more detailed explanations.

Table 3a shows results without VAD. The first column contains the results for which no chroma information was used ($b = 0.0$). The results in the second and third columns show that the additional information provided by the chord labels substantially boosts accuracy, for example from 38.4% ($a = 1.0$, $b = 0.0$) to the best result of 88.0% ($a = 1.0$, $b = 1.0$). While chord information yields the greatest improvement, we can also observe that using chord information alone does not provide a very good alignment result. For example, consider a chroma weight fixed at $b = 1.0$: when the phoneme feature is "off" ($a = 0.0$), we obtain only a mediocre alignment result of 59.1% accuracy, but setting $a = 1.0$, we obtain the top result of 88.0%. Our interpretation of these results is intuitive. Since chords occupy longer time spans and are therefore "hard to miss", they provide large scale alignment. The phonemes in the lyrics, on the other hand, often have very short time spans and are easy to miss in a rich polyphonic music environment. However, with good large scale alignment—as provided by the chords—their short term lyrics-to-audio alignment capabilities exceed those of chords.

We can also observe that when no chord information is given, VAD provides a similar kind of large scale alignment: for the baseline method ($a = 1.0$, $b = 0.0$) the use of VAD increases accuracy from 38.4% (Table 3a) to 46.0% (Table 3b). Table 3b also shows that using chroma and VAD together results in accuracy values slightly lower than the top ones, which has been caused by regions erroneously classified as non-vocal by VAD. The conclusion is: if full chord information is not available, use VAD; if chord information is available, use chroma instead.

The same pattern emerges for the mean absolute displacement d of words (Table 4). Here also, the best value, 0.51 seconds, is achieved when using both chroma and phoneme features (without VAD), compared to 1.26 seconds for the best result of the baseline method (with VAD).

One downside to the best methods mentioned so far is that the *complete chords* assumption is not always fulfilled, since transcribers often omit chord annotations for harmonically repeated sections (see Figure 2). The next section presents a method which—through intelligent use of the remaining chord information—deals with this situation and achieves results approaching the best ones seen above.

		chroma weight		
		$b = 0.0$	$b = 0.5$	$b = 1.0$
phoneme weight	$a = 0.0$	—	59.1	59.1
	$a = 0.5$	34.8	86.8	83.1
	$a = 1.0$	38.4	87.6	**88.0**

(a) accuracy without VAD

		chroma weight		
		$b = 0.0$	$b = 0.5$	$b = 1.0$
phoneme weight	$a = 0.0$	—	52.3	52.3
	$a = 0.5$	42.0	77.8	77.3
	$a = 1.0$	46.0	**81.9**	78.5

(b) accuracy with VAD

Table 3: **Accuracy:** mean percentage p as defined in Equation (2) for tests without and with vocal activity detection (VAD). Chroma and phoneme feature combined lead to the best results for the method using complete chord information. For a detailed discussion see Section 2.3.

		chroma weight		
		$b = 0.0$	$b = 0.5$	$b = 1.0$
phoneme weight	$a = 0.0$	—	1.98	1.99
	$a = 0.5$	8.22	0.63	1.06
	$a = 1.0$	6.93	0.72	**0.51**

(a) mean absolute displacement without VAD

		chroma weight		
		$b = 0.0$	$b = 0.5$	$b = 1.0$
phoneme weight	$a = 0.0$	—	3.74	3.73
	$a = 0.5$	5.52	1.67	1.69
	$a = 1.0$	4.67	**1.26**	1.65

(b) mean absolute displacement with VAD

Table 4: **Mean absolute displacement** d in seconds as defined in Equation (3) for tests without and with vocal activity detection (VAD). For a detailed discussion see Section 2.3.

3. RECOVERING PARTIALLY MISSING CHORDS

As we have seen in Figure 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 relaxing the "complete chords" assumption given in Section 2 and replace it with an assumption that is more in line with real world files:

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 Figure 2 now comply with our new set of assumptions.

While our basic method from Section 2 works in a single alignment step, the recovery method proposed in this section consists of the following three steps:

(a) naïve alignment: the basic alignment method as in Section 2, but missing chords are modelled by a "no chord" profile.

(b) phrase-level segmentation: the results of the alignment are used to build a new chord-based HMM for phrase-level segmentation.

(c) constrained alignment: the phrase-level segmentation result is fed back to the original alignment HMM: inference is performed constrained by phrase location.

Sections 3.1 to 3.3 will explain these steps in more detail and Section 3.4 presents the results.

3.1 Naïve Alignment

In this context we call "naïve" taking the best performing model from Section 2 ($a = 1.0$, $b = 1.0$, 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. This is step (a) in the above method outline. As could be expected, the scarcity of information leads to a substantial performance decrease, from 88.0% (as discussed in the previous section) to 58.44%. 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 2.3 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 of the "naïvely" obtained results to the song segments annotated with chords, we obtain a higher level of accuracy: 72.1%. This has motivated us to implement the following two steps (b) and (c).

3.2 Phrase-level Segmentation

This is step (b), according to the system overview given above. We build a new HMM based entirely on chords and chroma, with three hierarchical levels depicted in Figure 4: chord, song segment, and song, based on the first (naïve) alignment step explained above. We assume indeed that in segments with complete chord information the word time estimates are nearly correct. Since the words are associated with chords, they provide us with an estimate of the chord lengths for every segment type. For each segment with

complete chords we will 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 Figure 2, are duplicated such that a line always starts with a chord. This is important because otherwise the model based only on chroma observations could not find the correct new phrase beginning.

The model of each chord is determined by its length ℓ in seconds: it is modelled by $\lceil 2\ell \rceil$ states, i.e. two states per second. Only self-transitions or transitions to the next state are allowed (see Figure 4a). The self-transition probability is $s = 0.2$, and hence the expected duration of one state is 0.5 seconds at a frame rate of 10 Hz [5] . 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 modelled 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 [19]—in which the probability of very short chord durations is reduced.

The chord models are then concatenated to form a left-to-right model of the chord progression in one segment, as illustrated in Figure 4b. The segment HMMs are then 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 Figure 4c. Of course, segment models may appear several times.

3.3 Constrained Alignment

This third step (c) integrates the results calculated in the two previous steps (a) and (b). We now run the HMM inference from the first step once again, but using the estimated lyric line beginnings from the previous step (b): we constrain the Viterbi search at frames of line beginnings to exactly the word the line starts with. To be precise, it is the short pause state preceding this word that is fixed to start at the estimated line beginning, so that lines that start with a chord, but no lyrics are not forced to start with a word.

3.4 Results II

We chose the method that performed best in the experiments reported in Section 2.3 for the further experiments, i.e. the feature weight parameters are set to $a = 1.0$ and $b = 1.0$. We used the same eight songs (see Table 2) and performed two more experiments, firstly the naïve application (a) of the original chord and lyrics alignment method, and secondly the full method including steps (a), (b) and (c). The accuracy results are given in Figure 5, together with the result obtained under complete chord information. First of all, we observe that the results of the naïve method are worse than the results with complete chord information: with respect to the method with complete chords at the top of the figure, removing chord information clearly decreases accuracy (defined in Equation 2) from 88.0% to 58.4%. Our proposed method, i.e. steps (a) to (c), can recover much of the lost information by applying phrase constraints, resulting in an accuracy of 72.7%.

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

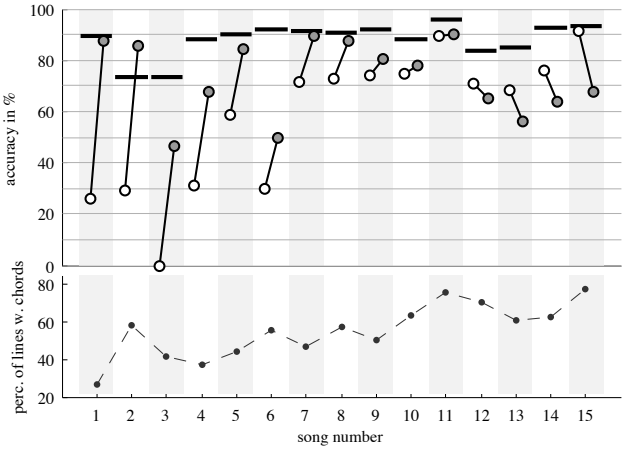


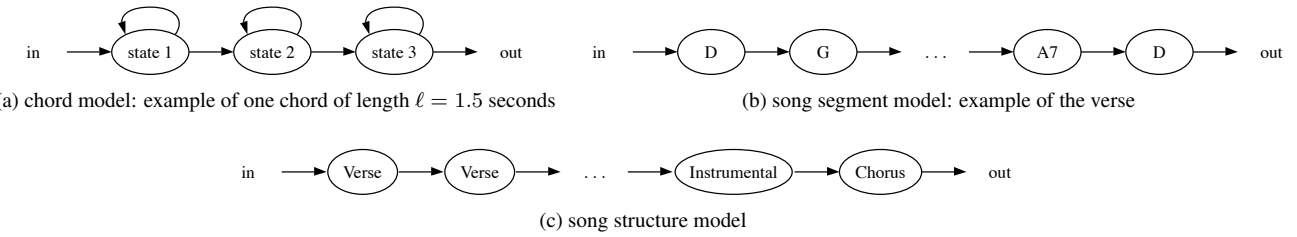
Figure 5: Songwise comparison. Top figure: black bars represent the accuracy obtained using all chord information (see Section 2); blank circles represent accuracy with partial chord data (chords of repeated segments removed); filled circles represent accuracy using our chord information recovery method as explained in Section 3. Bottom figure: proportion of lyrics lines for which chord information is available after partial removal.

method	accuracy in %
full chord information	88.0
incomplete chords (naïve method)	58.4
incomplete chords with recovery	72.7

Table 5: Accuracy for the methods presented in Section 3, as explained in 3.4.

Figure 5 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 naïve method (blank) to the method with chord information recovery (filled). The songs are sorted by amount of improvement (same order as given in Table 2), and we observe that the recovery method improves results for the first 11 (of 15) songs. The differences are more substantial, if the accuracy of the naïve method is low. It is also interesting to observe that the improvement correlates negatively with the amount of chord information provided (see the percentage of lines with available chord information at the bottom of Figure 5).

Our results imply furthermore that the segmentation achieved in step (b) is very good. 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 Figure 6 demonstrates.



(a) chord model: example of one chord of length $\ell = 1.5$ seconds

(b) song segment model: example of the verse

(c) song structure model

Figure 4: 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.

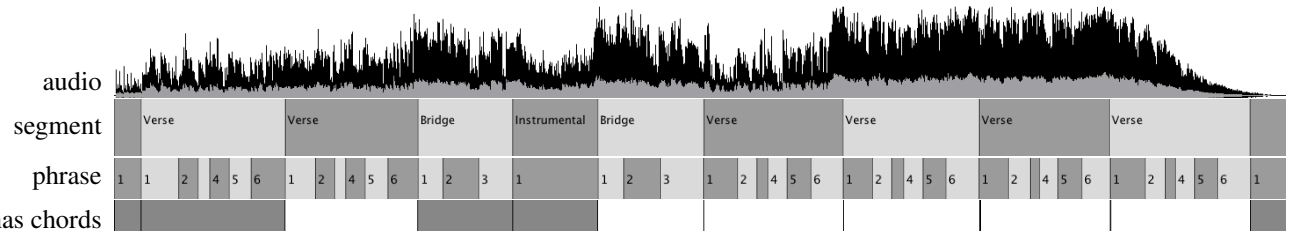


Figure 6: Automatic segmentation as explained in Section 3.2. The top line is a representation of the audio waveform of the song *Eternal Flame* by the Bangles, with means and maxima of positive values indicated in grey and black, respectively. Below are the automatically detected segments, with names 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.

4. DISCUSSION AND FUTURE WORK

Clearly, 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 all three steps into one would be much 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. Once found, such a model could also provide more accurate alignment for cases in which the chord labelling is of low quality, e.g. when chords are not written exactly over the right words.

In a more efficient system, the audio feature generation should be unified to avoid the overhead of operating with different sample rates. We also plan to implement an application of the methods presented here: a machine that automatically generates guitar/singing karaoke annotations and allows a new advanced karaoke experience for musicians. In fact, real-time alignment could make such an application an electronic lead-sheet tool for bands. 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. This is an interesting research problem in itself.

However, our primary goal is to further relax the assumptions made about the chord and lyrics data. For example, dropping the requirement that the succession or names of the song segments are given would make our method even more applicable to "sloppy" real world song annotations, and the segmentation method based on chord progressions presented in Section 3.3 is a promising starting point to finding song segments with no *a priori* information about their order. The next great music computing challenge is then to perform the task of chord-aided alignment without any prior chord information and automatically generate Internet-style chord and lyrics transcriptions. Though several systems for fully automatic structural segmentation and chord extraction exist, we are aware of none that combine the different parts needed: integrating more musical features is however one of the goals of the Sound and Music Computing Roadmap [6] and we expect that the most interesting music computing applications of the future will be those that aim to reach that goal.

5. CONCLUSIONS

This paper has shown that additional chord information in a textual "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

[6] `http://smcnetwork.org/roadmap`

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 MFCC 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 88.0% compared to 46.0% without chroma and 59.1% without phoneme features. If parts of the chord information are removed, the method performs worse (58.4%), 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 boosts accuracy by more than 14 percentage points to 72.7%. We show that the improvement on individual songs is particularly marked when large parts of the chord information are missing.

We have noted that the results of the second method imply a good performance of the segmentation method. 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.

6. ACKNOWLEDGEMENTS

We would like to thank Anssi Klapuri and Gaël Richard for their ideas. This research was supported by CrestMuse, CREST, JST.

7. REFERENCES

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

[2] F. Bronson, *The Billboard Book of Number One Hits*. Billboard Books, 1997.

[3] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 4, pp. 96–104, 2005.

[4] Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin, "LyricAlly: Automatic synchronization of acoustic musical signals and textual lyrics," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pp. 212–219, 2004.

[5] M.-Y. Kan, Y. Wang, D. Iskandar, T. L. Nwe, and A. Shenoy, "LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 338–349, 2008.

[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. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, 2008.

[7] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, 2008.

[8] C. H. Wong, W. M. Szeto, and K. H. Wong, "Automatic lyrics alignment for cantonese popular music," *Multimedia Systems*, vol. 12, no. 4–5, pp. 307–323, 2007.

[9] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical, and jazz music databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 287–288, 2002.

[10] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *to appear in IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[11] M. Goto, "A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA Journal)*, vol. 43, no. 4, pp. 311–329, 2004.

[12] T. Fujishima, "Real time chord recognition of musical sound: a system using Common Lisp Music," in *Proceedings of the International Computer Music Conference (ICMC 1999)*, pp. 464–467, 1999.

[13] L. Oudre, Y. Grenier, and C. Févotte, "Template-based chord recognition: Influence of the chord types," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pp. 153–158, 2009.

[14] A. Sheh and D. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.

[15] R. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proceedings of the International Computer Music Conference (ICMC 2003)*, pp. 27–34, 2003.

[16] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

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

[18] M. Mauch, *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary University of London, 2010.

[19] M. Mauch and S. Dixon, "A discrete mixture model for chord labelling," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pp. 45–50, 2008.