AUTOMATIC ALIGNMENT OF MUSIC AUDIO AND LYRICS

Annamaria Mesaros,
Dept. of Signal Processing
Tampere University of Technology
Tampere, Finland
annamaria.mesaros@tut.fi

*Tuomas Virtanen,**
Dept. of Signal Processing
Tampere University of Technology
Tampere, Finland
tuomas.virtanen@tut.fi

ABSTRACT

This paper proposes an algorithm for aligning singing in polyphonic music audio with textual lyrics. As preprocessing, the system uses a voice separation algorithm based on melody transcription and sinusoidal modeling. The alignment is based on a hidden Markov model speech recognizer where the acoustic model is adapted to singing voice. The textual input is preprocessed to create a language model consisting of a sequence of phonemes, pauses and possible instrumental breaks. Viterbi algorithm is used to align the audio features with the text. On a test set consisting of 17 commercial recordings, the system achieves an average absolute error of 1.40 seconds in aligning lines of the lyrics.

1. INTRODUCTION

Most of the work on music information retrieval (MIR) has been concentrated on acoustic information, i.e. an audio signal. Specific research problems include, for example, automatic music transcription, singer identification, and structure analysis. Also the lyrics are an important aspect in vocalized music since they convey most of the emotional part through semantic information. Early attempts to perform lyrics recognition using a large-vocabulary speech recognizer were successful only on pure singing voice [1, 2]. For commercial polyphonic recordings, the musical signals are significantly more complex than pure singing, thus phonetic models created for pure singing voice are more likely to fail. Also, the recognition for transcription is often not necessary, as the lyrics are easily found on the Internet.

This paper deals with the alignment of music audio with singing vocals and instrumental accompaniment with the corresponding textual lyrics, i.e., finding the temporal relationship between the two inputs. The alignment can be directly applied in automated karaoke annotation systems, but it has also potential in automatic singing database labeling and keyword spotting in singing database search algorithms. The problem can be viewed as an intermediate goal in the significantly harder problem of recognizing lyrics in polyphonic audio.

A straightforward way to do alignment is by creating a phonetic transcription of the word sequence comprising the text in the lyrics and aligning the phone sequence with the audio using a hidden Markov model (HMM) speech recognizer. For alignment, the possible paths in the Viterbi search algorithm of a speech recognizer are restricted to just one string of phonemes, representing the input text. However, there are significant differences in the dynamics and general properties of speech and singing sounds. The complexity of the polyphonic music signal compared to a pure singing voice signal is one more factor that makes the direct use of a speech recognizer difficult.

In this paper we present a system that automatically aligns a real-life piece of music to the corresponding textual lyrics. The proposed system consists of several processing steps. The audio signal is first preprocessed to separate the singing voice from the polyphonic signal. The alignment step employs a phonetic HMM recognizer to align the text in the lyrics to the singing.

The rest of this paper is organized as follows. We continue with a literature review of the related work. In Section 3 we present the overview of our system, in Section 4 the separation algorithm, and in Section 5 the models and rules for the alignment. Section 6 presents the experimental results, and Section 7 presents the conclusions of this work.

2. RELATED WORK

There is little work related to automatic alignment of lyrics text to singing in polyphonic music. Wong et al. [3] define the problem of aligning a music signal with singing in Cantonese to a corresponding lyric file. A preprocessing method is used to enhance the singing voice and to suppress background music. The choice of features for the alignment is based on the characteristics of Cantonese language. The system cannot handle non-tonal languages such as English.

The authors of [4] present a system based on Viterbi alignment for synchronizing lyrics with music CD recordings. The system uses a method for segregating vocals from a polyphonic music signal. Nonvocal regions are removed from the segregated signal by means of a vocal activity detection method. A language model is created for the forced alignment, by retaining only vowels for japanese lyrics converted to phonemes. The gender dependent monophone model of ISRC software is used as an initial phone model and is adapted to the singing voice characteristics. The alignment is done at phrase level, authors define a phrase as a section that was delimited by a space of line feed in the original lyrics. As a quality measure, the authors give alignment accuracy as the proportion of the length of the sections which are correctly labeled to the total length of a song, reporting 8 out of 10 songs with over 90% accuracy.

In [5], the authors present LyricAlly, a system that aligns first the higher level structure of a song and then within the boundaries of the detected sections, performs a line-level alignment. The line-level alignment uses only an uniform estimated phoneme duration, rather than a phoneme recognition based method. The system works by finding vocal segments but not recognizing their content. The line-level aligner will search for a target number of vocal segments, corresponding to the number of lines in the correspond-

* This work has been funded by the Academy of Finland

ing lyrics section. The method is based on assumptions about the structure and meter of the song and is limited to certain types of songs. The authors reported a 0.58 s mean and 3.6 s standard deviation for the error in line starting points alignment on a test set of 20 songs.

3. SYSTEM OVERVIEW

The audio file is preprocessed to obtain the vocal line. For this, we use a melody transcription system followed by a sinusoidal synthesis as described in Section 4. After extracting the vocal line, we extract features of the audio.

The alignment system is a HMM phonetic recognizer. It uses the 39 phonemes of the CMU pronouncing dictionary, plus short pause, silence and instrumental noise models. The monophone models are trained using a speech database. Furthermore, using maximum-likelihood linear regression (MLLR) speaker adaptation technique, the monophone models are adapted to clean singing voice characteristics. We use the Hidden Markov Model Toolkit (HTK) [6] for feature extraction, training and adaptation of the models and for the Viterbi alignment.

The text file contains the corresponding lyrics. We assume correct words and accurate representation of the sung words. The text is processed to obtain a sequence of words with optional silence, pause and noise between them. The transcription from words to phonemes is done using the CMU pronouncing dictionary.

4. VOCALS SEPARATION

The vocal separation is done in two stages where an automatic melody transcription algorithm is first used to estimate the notes of the the main vocal line and then sinusoidal modeling is used to represent and separate the corresponding acoustic signal. The transcription algorithm produces a robust mid-level representation of the pitch of the melody whereas the sinusoidal model allows estimating more accurate time-varying parameters.

4.1. Melody transcription

The melody transcription is done using the algorithm of Ryynänen and Klapuri [7], which takes polyphonic music signal as an input and estimates the note sequency corresponding to the lead-vocal melody. Each note is parameterized by its pitch, onset time, and duration. The transcription algorithm first measures the so-called salience of different fundamental frequencies at 23.2 ms intervals. The saliences are used as features for a hidden Markov model (HMM) consisting of a network of melody notes, background notes, and silence. Each melody or background note is modeled with a three-state left-to-right HMM, notes of different pitches having different parameters. The parameters are estimated from several hours of real music, for which the reference pitches were known.

The HMM network for modeling the whole signal is obtained by allowing transitions from the end states of notes to the beginning states of other notes, or to the silence state. Musicological information about the probability of different note sequences is incorporated by a bigram which defines the transitions between different notes. The transcription of the melody is obtained by finding the most likely melody note sequence using the Viterbi algorithm. In each frame, local maximum of the fundamental frequency salience in the vicinity of the transcribed note is used as a more accurate estimate of the pitch of the melody.

4.2. Sinusoidal modeling

The signal-level separation is based on the model $x(k) = v(k) + b(k)$ where the polyphonic input signal $x(k)$ is represented as the sum of vocal signal $v(k)$ and backround $b(k)$, k being the time index in samples. The vocal signal is further modeled using the sinusoidal model

$$v(k) = \sum_{n=1}^{N} a_n(k) \sin(\theta_n(k)) \tag{1}$$

where N is the number of overtones, and $a_n(k)$ and $\theta_n(k)$ are the amplitude and phase of the $n\mathrm{th}$ overtone at time k. In this study the number of overtones was fixed to 40.

In the estimation of the overtone parameters the signal is divided into 40 ms frames with 20 ms hop between adjacent frames. In each frame m a fixed amplitude $a_{m,n}$, frequency $f_{m,n}$, and phase $\theta_{m,n}$ is estimated for each overtone n as follows. The overtone frequencies are set to integer multiplies of the pitch of the melody, i.e. $f_{m,n} = nf_m$, where f_m is the estimated pitch in frame m. Complex correlation $c_{m,n}$ between the windowed signal and a sinusoid having the overtone frequency is calculated as

$$c_{m,n} = \frac{\sum_k x(k) \exp(i2\pi f_{m,n}/f_s) w(k)}{Z} \tag{2}$$

where f_s is the sampling frequency and $Z = \sum_k w(k)/2$ is a normalization constant. $w(k)$ is the Hamming window centered to the temporal position of each frame. The amplitude of the overtone is then obtained as the absolute value of the complex correlation and the phase by its angle. The estimation is repeated for each overtone in each frame.

Each overtone is synthesized by interpolating the parameters from frame to frame. Here the best quality was obtained using the method proposed in [8] which uses quadratic interpolation for the phases $\theta_n(k)$ and linear interpolation for the amplitudes $a_n(k)$. The vocal signal is obtained generating and summing the overtones according to (1).

5. ALIGNMENT

The alignment stage fits the sequence of words in the input text to the observed acoustic features of the separated vocal signal.

5.1. Lyrics processing

The lyrics input is transformed into a sequence of words that will be used by the recognizer. An optional short pause is inserted between each two words in the lyrics. At the end of each line we insert optional silence or noise event, to account for the voice rest and possible background accompaniment. An example of resulting recognition grammar for one of the test songs is:

[sil | noise] I [sp] BELIEVE [sp] I [sp] CAN [sp] FLY [sil | noise] I [sp] BELIEVE [sp] I [sp] CAN [sp] TOUCH [sp] THE [sp] SKY [sil | noise] I [sp] THINK [sp] ABOUT [sp] IT [sp] EVERY [sp] NIGHT [sp] AND [sp] DAY [sil | noise] SPREAD [sp] MY

[sp] WINGS [sp] AND [sp] FLY [sp] AWAY [sil | noise]
where the [ ] encloses options and | denotes alternatives. This way, the alignment algorithm can choose to include pauses and noise where needed.

The phonetic transcription of the recognition grammar is obtained using the CMU pronouncing dictionary. The features extracted from the separated vocals are aligned with the obtained string of phonemes, using the Viterbi forced alignment.

5.2. The set of models

As features, we used 13 mel-frequency cepstral coefficients plus delta and acceleration coefficients calculated on 25 ms frames with a 10 ms hop between adjacent frames. A left-to-right HMM with 3 states is used to represent each phoneme. The silence model is a fully-conected HMM with 3 states and the short pause model is a one-state HMM. An additional model for the instrumental noise was used, accounting for the distorted instrumental regions that can appear in the separated vocals signal. The noise model is a 5-states fully connected HMM.

In absence of an annotated dabase of singing phonemes, the set of monophone models was trained using the entire ARCTIC speech database [1]. Silence and short pause models were trained on the same material. The noise model was separately trained on instrumental sections from different songs, others than the ones in the test database.

5.3. Model adaptation

In speech recognition, a speaker independent model set can be adapted to fit the characteristics of an individual speaker by using some adaptation technique. Similar to speaker adaptation, we can use smaller amount of training data to adapt HMM models trained on speech to better match the characteristics of singing voice.

Maximum likelihood linear regression (MLLR) computes a set of transformations that will reduce the mismatch between an initial model and the adaptation data. Specifically, the method estimates a set of linear transformations for the mean and variance of a Gaussian mixture HMM system [9]. The effect of these transformations is to shift the component means and alterate the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data. We only used adaptation of the mean.

The transformation matrix used to give a new estimate of the adapted mean is given by

$$\hat{\mu} = \mathbf{W}\xi \tag{3}$$

where $\mathbf{W}$ is the $n \times (n+1)$ transformation matrix (n is the dimensionality of the data). ξ is the extended mean vector $\xi = \begin{bmatrix} 1 & \mu_1 & \mu_2 & \mu_3 & \ldots & \mu_n \end{bmatrix}^T$.

The transformation matrix is obtained by solving a maximization problem using the Expectation-Maximization technique. The adaptation method is implemented in the HTK [6].

If the true transcription of the adaptation data is known, then the adaptation can be done in a supervised way. A number of 49 monophonic singing fragments ranging from 20 to 30 seconds and their transcriptions were used for adapting the phone models to singing voice. We used a singing database comprising fragments of pop songs. The adaptation data consists in 19 male and 30 female voice fragments, sung by 12 female and 7 male nonprofessional singers. Following the supervised MLLR speaker adaptation, the method was used to obtain a singing voice adapted set of 39 phones. These models will be used in the alignment process.

6. SYSTEM EVALUATION AND DISCUSSION

A number of 17 songs was chosen at random from a large structure annotated database of commercial popular music. The alignment system processes text and music representing a vocal section of a song. We define a vocal section as being the verse, chorus, or bridge, which contains vocals and instrumental accompaniment. The sections were manually annotated for reference. The section lengths range from 9 to 40 seconds.

The lyrics were retrieved from public lyrics database and were manually checked for correctness. The text input contains a number of lines, each line corresponding roughly to one singing phrase. Each vocal section of each song was manually paired with the corresponding lyrics, resulting in a number of 100 pairs of audio and text to be aligned. The number of lyric lines per each section was between 3 and 9.

The output of the alignment system consists in time information about the limits of each text line. As a performance measure of the alignment we use the mean of the absolute alignment errors in seconds at the beginning and at the end of each lyric line. The distribution of the alignment errors is presented in Figure 2. It has a mean of 0.12 and standard deviation of 2.23 seconds. The mean absolute error for the entire test set is 1.40 seconds, and the median is 0.64 seconds.

A manual analysis of the errors shows that one main reason for misalignments is a faulty output of the vocal separation stage. Some of the songs are from pop-rock genre, featuring loud instruments as an accompaniment, and the melody transcription fails to pick the voice signal. In this case, the output contains a mixture of the vocals with some instrumental sounds, but the voice is usually too distorted to be recognizable. In other cases, there are inaccuracies in the annotated temporal locations of the lines, where singing goes from one line directly into the other with no pause. In these cases, even the manual annotation can have ambiguity.

Overall, the system works very well with music where the breathing pauses in singing are at the end of the lyric lines and the vocals are strong compared to the accompaniment.

The error in seconds is not an ideal measure, as it can be perceptually different for songs with different tempos, but the system uses only phonetic information for the alignment. One possible refinement is by using musical knowledge to synchronize the lyric lines to the musical bars.

Table 1: *Experimental results: alignment errors*

mean error	0.12 s
mean absolute error	1.40 s

[1] http://festvox.org/cmu_arctic/

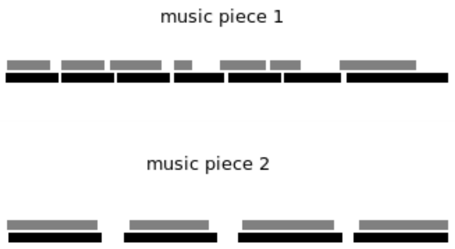

Figure 1: *Experimental results: automatic alignment examples. Black line - manual annotation, grey line - alignment system output*

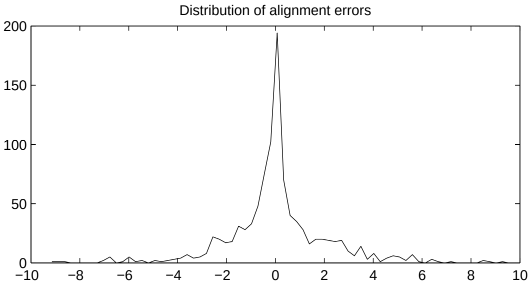

Figure 2: *Experimental results: Distribution of alignment errors*

7. CONCLUSIONS

This paper presented a system for automatic alignment of music audio and lyrics for the vocal sections of a song. The system uses an HMM speech recognizer based on monophone models and a voice separation preprocessing of the audio. The alignment grammar is constructed from the lyrics text by allowing optional pauses after each word, and instrumental noise between each two lines in the lyrics. Experimental results on a set of commercial recordings were satisfactory, half of the test material was aligned within 0.64 seconds absolute error. For the entire test set, an average absolute error of 1.40 seconds was obtained.

8. REFERENCES

[1] A. Loscos, P. Cano, and J. Bonada, "Low-delay singing voice alignment to text," in *Proceedings of the International Computer Music Conference (ICMC)*, 1999.

[2] T. Hosoya, M. Suzuki, A. Ito, and S. Makino, "Lyrics recognition from a singing voice based on finite state automaton for music information retrieval," in *Proceedings of the 6th International Conference on Music Information Retrieval ISMIR*, 2005.

[3] Chi Hang Wong, Wai Man Szeto, and Kin Hong Wong, "Automatic lyrics alignment for Cantonese popular music," *Multimedia Systems*, vol. 12, no. 4-5, 2007.

[4] Hiromasa Fujihara, Masataka Goto, Jun Ogata, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals," in *ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia*, Washington, DC, USA, 2006.

[5] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin, "LyricAlly: automatic synchronization of acoustic musical signals and textual lyrics," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, 2004.

[6] "Cambridge University Engineering Department. The Hidden Markov Model Toolkit (HTK)," http://htk.eng.cam.ac.uk/.

[7] Matti Ryynänen and Anssi Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, accepted for publication.

[8] Y. Ding and X. Qian, "Processing of musical tones using combined quadratic polynomial-phase sinusoid and residual (QUASAR) signal model," *Journal of the Audio Engineering Society*, vol. 45, no. 7/8, 1997.

[9] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Language*, vol. 10, 1996.