Combining Pitch-Based Inference and Non-Negative Spectrogram Factorization in Separating Vocals from Polyphonic Music

Tuomas Virtanen, Annamaria Mesaros, Matti Ryynänen

Department of Signal Processing, Tampere University of Technology, Finland

tuomas.virtanen@tut.fi, annamaria.mesaros@tut.fi, matti.ryynanen@tut.fi

Abstract

This paper proposes a novel algorithm for separating vocals from polyphonic music accompaniment. Based on pitch estimation, the method first creates a binary mask indicating time-frequency segments in the magnitude spectrogram where harmonic content of the vocal signal is present. Second, non-negative matrix factorization (NMF) is applied on the non-vocal segments of the spectrogram in order to learn a model for the accompaniment. NMF predicts the amount of noise in the vocal segments, which allows separating vocals and noise even when they overlap in time and frequency. Simulations with commercial and synthesized acoustic material show an average improvement of 1.3 dB and 1.8 dB, respectively, in comparison with a reference algorithm based on sinusoidal modeling, and also the perceptual quality of the separated vocals is clearly improved. The method was also tested in aligning separated vocals and textual lyrics, where it produced better results than the reference method.

Index Terms: sound source separation, non-negative matrix factorization, unsupervised learning, pitch estimation

1. Introduction

Separation of sound sources is a key phase in many audio analysis tasks since real-world acoustic recordings often contain multiple sound sources. Humans are extremely skillful in "hearing out" the individual sources in the acoustic mixture. A similar ability is usually required in computational analysis of acoustic mixtures. For example in automatic speech recognition, additive interference has turned out to be one of the major limitations in the existing recognition algorithms.

A significant amount of existing monaural (one-channel) source separation algorithms are based on either pitch-based inference or spectrogram factorization techniques. Pitch-based inference algorithms (see Section 2.1 for a short review) utilize the harmonic structure of sounds, estimate the time-varying fundamental frequencies of sounds, and apply this in the separation. Spectrogram factorization techniques (see Section 2.2), on the other hand, utilize the redundancy of the sources by decomposing the input signal into a sum of repetitive components, and then assign each component to a sound source.

This paper proposes a hybrid system where pitch-based inference is combined with unsupervised spectrogram factorization in order to achieve a better separation quality of vocal signals in accompanying polyphonic music. The hybrid system proposed in Section 3 first estimates the fundamental frequency of the vocal signal. Then a binary mask is generated which covers time-frequency regions where the vocal signals are present. A non-negative spectrogram factorization algorithm is applied on the non-vocal regions. This stage produces an estimate of the contribution of the accompaniment in the vocal regions of the spectrogram using the redundancy in accompanying sources. The estimated accompaniment can then be subtracted to achieve better separation quality, as shown in the simulations in Section 4. The proposed system was also tested in aligning separated vocals with textual lyrics, where it produced better results than the previous algorithm, as explained in Section 5.

2. Background

Majority of the existing sound source separation algorithms are based either on pitch-based inference or spectrogram factorization techniques, both of which are shortly reviewed in the following two subsections.

2.1. Pitch-based inference

Voiced vocal signals and pitched musical instrument are roughly harmonic, which means that they consist of harmonic partials at approximately integer multiples of the fundamental frequency f_0 of the sound. An efficient model for these sounds is the sinusoidal model, where each partial is represented with a sinusoid with time-varying frequency, amplitude and phase.

There are many algorithms for estimating the sinusoidal modeling parameters. A robust approach is to first estimate the time-varying fundamental frequency of the target sound and then to use the estimate in obtaining more accurate parameters of each partial. The target vocal signal can be assumed to have the most prominent harmonic structure in the mixture signal, and there are algorithms for estimating the most prominent fundamental frequency over time, for example [1] and [2]. Partial frequencies can be assumed to be integer multiples of the fundamental frequency, but for example Fujihara et al. [3] improved the estimates by setting local maxima of the power spectrum around the initial partial frequency estimates to be the exact partial frequencies. Partial amplitudes and phases can then be estimated for example by picking the corresponding values from the amplitude and phase spectra.

Once the frequency, amplitude, and phase have been estimated for each partial in each frame, they can be interpolated to produce smooth amplitude and phase trajectories over time. For example, Fujihara et al. [3] used quadratic interpolation of phases. Finally the sinusoids can be generated and summed to produce an estimate of the vocal signal.

The above procedure produces good results especially when the accompanying sources do not have significant amount of energy at the partial frequencies. A drawback in the above procedure is that it assigns all the energy at partial frequencies to the target source. Especially in the case of music signals, sound sources are likely to appear in harmonic relationships so that many of the partials have the same frequency. Furthermore,

unpitched sounds may have a significant amount of energy at high frequencies, some of which overlaps with the partial frequencies of the target vocals. This causes the partial amplitudes to be overestimated and distorts the spectrum of separated vocal signal. The phenomenon has been addressed for example by Goto [2] who used prior distributions for the vocal spectra.

2.2. Spectrogram factorization

Recently, spectrogram factorization techniques such as non-negative matrix factorization (NMF) and its extensions have produced good results in sound source separation [4]. The algorithms employ the redundancy of the sources over time: by decomposing the signal into a sum of repetitive spectral components they lead to a representation where each sound source is represented with a distinct set of components.

The algorithms typically operate on a phase-invariant time-frequency representation such as the magnitude spectrogram. We denote the magnitude spectrogram of the input signal by $\mathbf{X}$, and its entries by $\mathbf{X}_{k,m}$, where $k = 1, \ldots, K$ is the discrete frequency index and $m = 1, \ldots, M$ is the frame index. In NMF the spectrogram is approximated as a product of two element-wise non-negative matrices, $\mathbf{X} \approx \mathbf{SA}$, where the columns of matrix $\mathbf{S}$ contain the spectra of components and the rows of matrix $\mathbf{A}$ their gains in each frame. $\mathbf{S}$ and $\mathbf{A}$ can be efficiently estimated by minimizing a chosen error criterion between $\mathbf{X}$ and the product $\mathbf{SA}$, while restricting their entries to non-negative values. A commonly used criterion is the divergence

$$D(\mathbf{X}||\mathbf{SA}) = \sum_{k=1}^{K} \sum_{m=1}^{M} d(\mathbf{X}_{k,m}, [\mathbf{SA}]_{k,m}) \qquad (1)$$

where the divergence function d is defined as

$$d(p, q) = p \log(p/q) - p + q. \qquad (2)$$

Once the components have been learned, those corresponding to the target source can be detected and further analyzed. A problem in the above method is that it is only capable of learning and separating redundant spectra in the mixture. If a part of the target sound is present only once in the mixture, it is unlikely to be well separated.

In comparison with the accompaniment in music, vocal signals have typically more diverse spectra. The fine structure of the short-time spectrum of a vocal signal is determined by its fundamental frequency and the rough shape of the spectrum is determined by the phonemes, i.e, sung words. In practice both of these vary as a function of time. Especially when the input signal is short, the above properties make learning of all the spectral components of the vocal signal a difficult task.

The above problem has been addressed for example by Raj et al. [5], who trained a set of spectra for the accompaniment using non-vocal segments which were manually annotated. Spectra of the vocal part was then learned from the mixture by keeping the accompaniment spectra fixed. Slightly similar approach was used by Ozerov et al. [6] who segmented the signal to vocal and non-vocal segments, and then a priorly trained background model was adapted using the non-vocal segments. The above methods require temporal non-vocal segments where the accompaniment is present without the vocals.

3. Proposed hybrid method

To overcome the limitations in the pitch-based and unsupervised learning approaches, we propose a hybrid system which utilizes the advantages of the both approaches. The block diagram of the system is presented in Figure 1. In the right processing branch, pitch-based inference and a binary mask is first used to identify time-frequency regions where the vocal signal is present, as explained in Section 3.1. Non-negative matrix factorization is then applied on the remaining non-vocal regions in order to learn an accompaniment model, as explained in Section 3.2. This stage also predicts the spectrogram of the accompanying sounds on the vocal segments. The predicted accompaniment is then subtracted from the vocal spectrogram regions, and the remaining spectrogram is inverted to get an estimate of the time-domain vocal signal, as explained in Section 3.3.

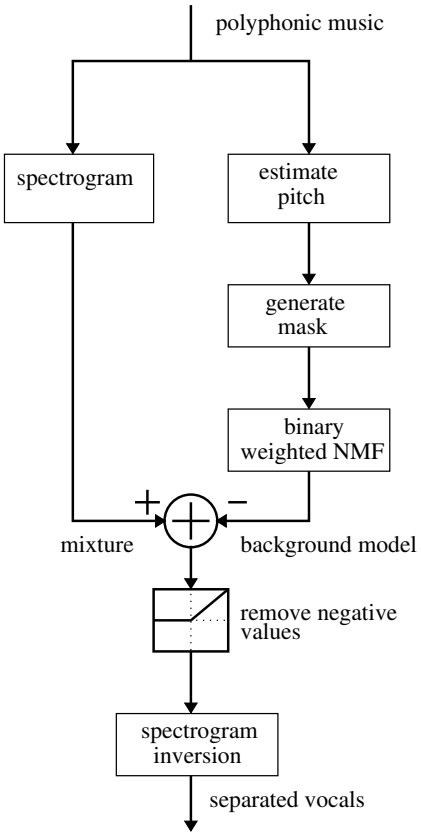

Figure 1: *The block diagram of the proposed system. See the text for an explanation.*

3.1. Pitch-based binary mask

A pitch estimator is first used to find the time-varying pitch of vocals in the input signal. Our main target in this work is music signals, and we found that the melody transcription algorithm of Ryynänen and Klapuri [7] produced good results in the pitch estimation. To get an accurate estimate of time-varying pitches, local maxima in the fundamental frequency salience function [7] around the quantized pitch values were interpreted as the exact pitches. The algorithm produces a pitch estimate at each 20 ms interval.

Based on the estimated pitch, time-frequency regions of the vocals are predicted. The accuracy of the pitch estimation algorithm was found to be good enough so that the partial frequencies were assigned to be exactly integer multiples of the estimated pitch. The NMF operates on the magnitude spectro-

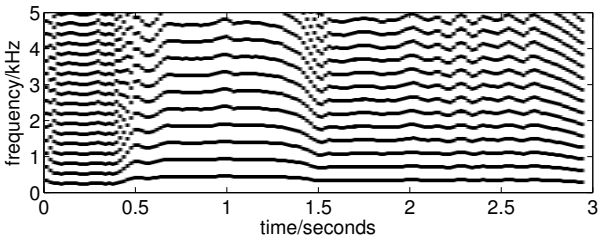

Figure 2: *An example of estimated vocal binary mask. Black color indicates vocal regions.*

gram obtained by short-time discrete Fourier transform (DFT), where DFT length is equal to N, the number of samples in each frame. Thus, the frequency axis of the spectrogram consist of a discrete set of frequencies $f_s k/N$, where $k = 0, \ldots, N/2$, since frequencies are used only up to the Nyquist frequency. In each frame, a fixed frequency region around each predicted partial frequency is then marked as a vocal region. In our system, a 50 Hz bandwidth around the predicted partial frequencies f was marked as the vocal region, meaning that if the frequency bin was within the 50 Hz interval, it was marked as the vocal region. On $N = 1764$, this leads to two or three frequency bins around the partial frequency marked as vocal segment, depending on the alignment between the partial frequency and the discrete frequency axis. In practice, a good bandwidth around each partial depends at least on the window length, which was 40 ms in our implementation. The pitch estimation stage can also produce an estimate of voice activity. For unvoiced frames all the frequency bins are marked as non-vocal regions.

Once the above procedure is applied in each frame, we obtain a K-by-M binary mask $\mathbf{W}$ where each entry indicates the vocal activity (0=vocals, 1=no vocals). An example of a binary mask is illustrated in Figure 2.

3.2. Binary weighted non-negative matrix factorization

A noise model is trained on non-vocal time-frequency segments corresponding to value 1 in the binary mask. The noise model is the same as in NMF, so that the magnitude spectrogram of noise is the product of a spectrum matrix $\mathbf{S}$ and gain matrix $\mathbf{A}$. The model is estimated by minimizing the divergence between the observed spectrogram $\mathbf{X}$ and the model $\mathbf{SA}$. Vocal regions (binary mask value 0) are ignored in the estimation, i.e., the error between $\mathbf{X}$ and $\mathbf{SA}$ is not measured on them. The above procedure allows using information of non-vocal time-frequency regions even in temporal segments where the vocals are present. Non-vocal regions occurring within a vocal segment enable predicting the accompaniment spectrogram for the vocal regions as well.

The background model is learned by minimizing the weighted divergence

$$D_{\mathbf{W}}(\mathbf{X}||\mathbf{SA}) = \sum_{k=1}^{K}\sum_{m=1}^{M} \mathbf{W}_{k,m} d(\mathbf{X}_{k,m}, [\mathbf{SA}]_{k,m}) \quad (3)$$

which is equivalent to

$$D_{\mathbf{W}}(\mathbf{X}||\mathbf{SA}) = D(\mathbf{W} \otimes \mathbf{X}||\mathbf{W} \otimes (\mathbf{SA})) \quad (4)$$

where $\otimes$ is element-wise multiplication.

The weighted divergence can be minimized by initializing $\mathbf{S}$ and $\mathbf{A}$ with random positive values, and then applying the following multiplicative update rules sequentially:

$$\mathbf{S} \leftarrow \mathbf{S} \otimes \frac{(\mathbf{W} \otimes \mathbf{X} \oslash \mathbf{SA})\mathbf{A}^{\mathsf{T}}}{\mathbf{W}\mathbf{A}^{\mathsf{T}}} \quad (5)$$

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\mathbf{S}^{\mathsf{T}}(\mathbf{W} \otimes \mathbf{X} \oslash \mathbf{SA})}{\mathbf{S}^{\mathsf{T}}\mathbf{W}} \quad (6)$$

Here both $\oslash$ and $\frac{\mathbf{X}}{\mathbf{Y}}$ denote element-wise division. The updates can be applied until the algorithm converges. In our studies 30 iterations was found to be sufficient for a good separation quality.

The convergence of the approach can be proved as follows. Let us write the weighted divergence in the form

$$D(\mathbf{W} \otimes \mathbf{X}||\mathbf{W} \otimes (\mathbf{SA})) = \sum_{m=1}^{M} D(\mathbf{W}_m \mathbf{x}_m||\mathbf{W}_m \mathbf{S}\mathbf{a}_m) \quad (7)$$

where $\mathbf{W}_m$ is a diagonal matrix where the elements of the mth column of $\mathbf{W}$ are on the diagonal, and $\mathbf{x}_m$ and $\mathbf{a}_m$ are the mth columns of matrices $\mathbf{X}$ and $\mathbf{A}$, respectively.

In the sum (7) the divergence of a frame is independent of other frames and the gains affect only individual frames. Therefore, we can derive the update for gains in individual frames. The right side of Eq. (7) can be expressed for an individual frame m as

$$D(\mathbf{W}_m \mathbf{x}_m||\mathbf{W}_m \mathbf{S}\mathbf{a}_m) = D(\mathbf{y}_m||\mathbf{B}_m \mathbf{a}_m) \quad (8)$$

where $\mathbf{y}_m = \mathbf{W}_m \mathbf{x}_m$ and $\mathbf{B}_m = \mathbf{W}_m \mathbf{S}$. For the above expression we can directly apply the update rule of Lee and Seung [8] which is given as

$$\mathbf{a}_m \leftarrow \mathbf{a}_m \otimes \frac{\mathbf{B}_m^{\mathsf{T}}(\mathbf{y}_m \oslash (\mathbf{B}_m \mathbf{a}_m))}{\mathbf{B}_m^{\mathsf{T}}\mathbf{1}} \quad (9)$$

where $\mathbf{1}$ is a all-one K-by-1 vector. The divergence (8) has been proved to be non-increasing under the update rule (9) by Lee and Seung [8]. By substituting $\mathbf{y}_m = \mathbf{W}_m \mathbf{x}_m$ and $\mathbf{B}_m = \mathbf{W}_m \mathbf{S}$ back to Eq. (9) we obtain

$$\mathbf{a}_m \leftarrow \mathbf{a}_m \otimes \frac{\mathbf{S}^{\mathsf{T}}\mathbf{W}_m(\mathbf{x}_m \oslash (\mathbf{S}\mathbf{a}_m))}{\mathbf{S}^{\mathsf{T}}\mathbf{W}_m} \quad (10)$$

The above equals (6) for each column of $\mathbf{A}$, and therefore the weighted divergence (3) is non-increasing under the update (6). The update rule (5) can be obtained similarly by changing the role of $\mathbf{S}$ and $\mathbf{A}$ by writing the weighted divergence using transposes of matrices as

$$D_{\mathbf{W}}(\mathbf{X}||\mathbf{SA}) = D_{\mathbf{W}^{\mathsf{T}}}(\mathbf{X}^{\mathsf{T}}||\mathbf{A}^{\mathsf{T}}\mathbf{S}^{\mathsf{T}}) \quad (11)$$

and following the above proof.

3.3. Vocal spectrogram inversion

The magnitude spectrogram $\mathbf{V}$ of vocals is reconstructed as

$$\mathbf{V} = [\max(\mathbf{X} - \mathbf{SA}, 0)] \otimes (\mathbf{1} - \mathbf{W}), \quad (12)$$

where $\mathbf{1}$ is K-by-M matrix which all entries equal 1. The operation $\mathbf{X} - \mathbf{SA}$ subtracts the estimated background from the observed mixture, and it was found advantageous to restrict this value above zero by the element-wise maximum operation. Element-wise multiplication by $(\mathbf{1} - \mathbf{W})$ allows non-zero magnitude only in the estimated vocal regions. The magnitude spectrogram of the background signal can be obtained as $\mathbf{X} - \mathbf{V}$.

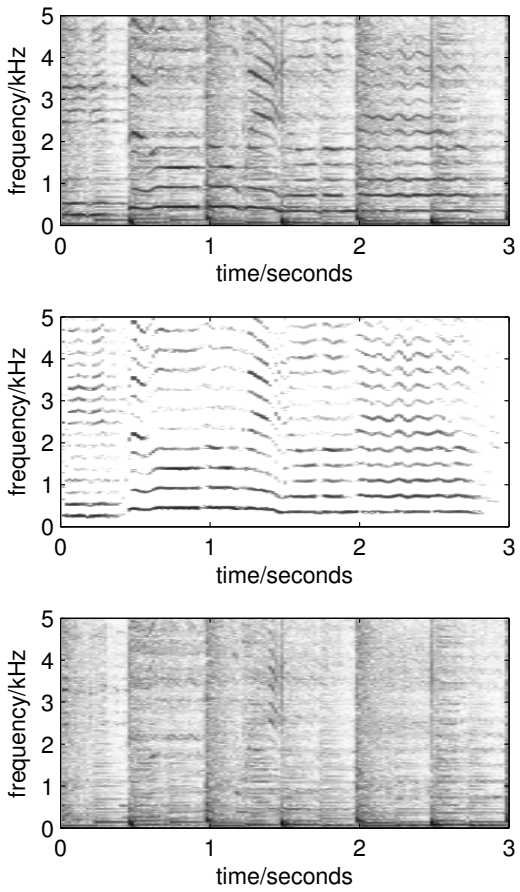

Figure 3: *Spectrograms of a polyphonic example mixture signal (top), separated vocals (middle) and separated accompaniment (bottom). The darker the color, the larger the magnitude at a certain time-frequency point.*

Figure 3 shows example spectrograms of a polyphonic signal, its separated vocals and background. Time-varying harmonic combs corresponding to voiced parts of the vocals present in the mixture signal are mostly removed from the estimated background.

Complex spectrogram is obtained by using the phases of the original mixture spectrogram, and finally the time-domain vocal signal can be obtained by overlap-add. Examples of separated vocal signals are available at `http://www.cs.tut.fi/~tuomasv/demopage.html`.

3.4. Discussion

We tested the method with various number of components (the number of columns in matrix $\mathbf{S}$). Depending on the length and complexity of the input signal, good results were obtained with a relatively small number of components (between 10 and 20) and iterations (10-30). However, the method does not seem to be very sensitive for the exact values of these parameters. On the other hand, we observed that a large number of components and iterations may lead to lower separation quality than fewer components and iterations. This is caused either by overfitting the accompaniment model or by learning undetected parts of the vocals by the accompaniment model. The above is substantially affected by the structure of the binary mask: a small number of bins in a frame marked as vocals is likely to reduce the quality. More detailed analysis of an optimal binary mask and NMF parameters is a topic for further research.

With a small number of iterations the proposed method is relatively fast and the total computation time is less than the length of the input signal on a 1.9 GHz desktop computer.

In addition to NMF, also more complex models (for example which allow time-varying spectra, see [9, 10]) can be used with the binary weight matrix, but in practice the NMF model was found to be sufficient. The model can also be extended so that the spectra for vocal parts can be learned from the data (as for example in [5]), but this requires relatively long input signal so that each pitch/phoneme combination is present in the signal multiple times.

4. Simulations

The performance of the proposed hybrid method was quantitatively evaluated using two sets of music signals. The first test set included 65 singing performances consisting of approximately 38 minutes of audio. For each performance, the vocal signal was mixed with a musical accompaniment signal to obtain a mixture signal, where the accompaniment signal was synthesized from the corresponding MIDI-accompaniment file. The signal levels were adjusted so that vocals-to-accompaniment ratio was -5 dB for each performance.

The second test set consisted of excerpts from nine songs on a karaoke DVD (Finnkidz 1, Svenska Karaokefabriken Ab, 2004). The DVD contains an accompaniment version of each song and also a version with lead vocals. The two versions are temporally synchronous at audio sample level so that the vocal signal could be obtained for evaluation by subtracting the accompaniment version from the lead-vocal version. The segments which include several simultaneous vocal signals (e.g., doubled vocal harmonies), were manually annotated in the songs and excluded from the evaluation. This resulted in approximately twenty minutes of audio, where the segment lengths varied from ten seconds to several minutes. The average relative ratio of the vocals and accompaniment in the DVD database was -4.0 dB.

Each segment was processed using the proposed method and also the below reference methods. All the methods use identical melody transcription algorithm, the one proposed by Ryynänen and Klapuri [7]. All the algorithms use 40 ms window size and 50% overlap between adjacent windows. The number of harmonic partials in all the methods was set to 60, and they used an identical binary mask. The number of NMF components was 20 and the number of iterations 30.

- Sinusoidal modeling. In the sinusoidal modeling algorithm the amplitude and phase were estimated by calculating the cross-correlation between the windowed and a complex exponential having the partial frequency. Quadratic interpolation of phases and linear interpolation of amplitudes was used in synthesizing the sinusoids.
- Binary masking does not subtract the background model subtraction but obtained the vocal spectrogram as: $\mathbf{V} = \mathbf{X} \otimes (\mathbf{1} - \mathbf{W})$
- The proposed method was also tested without vocal mask multiplication after the background model subtraction. In this method the vocal spectrogram was obtained as $\mathbf{V} = \max(\mathbf{X} - \mathbf{SA}, 0)$, and the method is denoted as "proposed*".

Table 1: *Average vocal-to-accompaniment ratio of the tested methods in dB.*

	data set	
method	set 1 (synthesized)	set 2 (Karaoke DVD)
proposed	2.1 dB	4.9 dB
sinusoidal	0.3 dB	3.6 dB
binary mask	-0.8 dB	2.9 dB
proposed*	2.1 dB	4.6 dB

The quality of the separation was measured by calculating the vocal-to-accompaniment ratio

$$\mathrm{VAR[dB]} = 10 \log_{10} \frac{\sum_n s(n)^2}{\sum_n (s(n) - \hat{s}(n))^2}, \qquad (13)$$

of each segment, where $s(n)$ is the reference vocal signal and $\hat{s}(n)$ is the separated vocal signal. The weighted average of VAR was calculated over the whole database by using the duration of each segment as its weight. Table 1 shows the results for both data sets and methods.

The results show that the proposed method achieves clearly better separation quality than the sinusoidal modeling and binary mask reference methods. All the methods are able to improve clearly the vocal-to-accompaniment ratio of the mixture signal, which were -5.0 dB and -4.0 dB for sets 1 and 2, respectively. Listening to the separated samples revealed that most of the errors, especially on the synthesized database, arise from errors on the transcription. The perceived quality of the separated vocals was significantly better with the proposed method than with the reference methods. The performance of the proposed* method is equal on set 1 and slightly worse on set 2, which shows that multiplication by the binary mask after subtracting the background model increases the quality slightly.

5. Application to audio and text alignment

One practical application for the vocal separation system is automatic alignment of a piece of music to the corresponding textual lyrics. Having a separated vocal signal allows the use of a phonetic hidden Markov model (HMM) recognizer to align the vocals to the text in the lyrics, similarly to text-to-speech alignment. A similar approach has been presented by Fujihara et al. in [3]. The system uses a method for segregating vocals from a polyphonic music signal, then a vocal activity detection method to remove the nonvocal regions. The language model is created by retaining only the vowels for Japanese lyrics converted to phonemes. As a refinement, in [11] Fujihara and Goto include a fricative detection for the /SH/ phoneme and a filler model consisting of vowels between consecutive phrases.

The language model in our alignment system consists of the 39 phonemes of the CMU pronouncing dictionary, plus short pause, silence, and instrumental noise models. The system does not use any vocal detection method, considering that the noise model is able to deal with the nonvocal regions. 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. Each monophone model was represented by a left-to-right HMM with 3 states. 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 was a 5-state fully-connected HMM. The emission distributions of the states were 20-component Gaussian mixture models (GMMs) for the monophone states and 5-component GMMs for the noise states.

In the absence of an annotated database of singing phonemes, the monophone models were trained using the entire ARCTIC speech database. 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. Furthermore, using maximum-likelihood linear regression (MLLR) speaker adaptation technique, the monophone models were adapted to clean singing voice characteristics using 49 monophonic singing fragments of popular music, their lengths ranging from 20 to 30 seconds.

The recognition grammar is determined by the sequence of words in the lyrics text file. The text is processed to obtain a sequence of words with optional short pause (sp) inserted between each two words and optional silence (sil) or noise at the end of each lyrics line, to account for the voice rest and possible accompaniment present in the separated vocals. A fragment of the resulting recognition grammar for an example piece of music 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]

where [ ] 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 was obtained using the CMU pronouncing dictionary. The features extracted from the separated vocals were aligned with the obtained string of phonemes, using the Viterbi forced alignment. The Hidden Markov Model Toolkit (HTK) [12] was used for feature extraction, training and adaptation of the models and for the Viterbi alignment.

Seventeen pieces of commercial popular music were used as test material. The alignment system processes text and music of manually annotated verse and chorus sections of the pieces. One hundred such sections with lengths ranging from 9 to 40 seconds were paired with corresponding lyrics text files. The timing of the lyrics was manually annotated for a reference.

In testing, the alignment system was used to align the separated vocals of a section with the corresponding text. As a performance measure of the alignment, we use the mean absolute alignment error in seconds at the beginning and at the end of each line in the lyrics.

We tested both the proposed method and the reference sinusoidal modeling algorithm, for which the mean absolute alignment errors were 1.33 and 1.37, respectively. Even though the difference is not large, this study shows that the proposed method enables more accurate information retrieval of vocal signals than the previous method.

6. Conclusions

We have proposed a novel algorithm for separating vocals from polyphonic music accompaniment. The method combines two powerful approaches, pitch-based inference and unsupervised non-negative matrix factorization. Using pitch estimate of the vocal signal, the method is able to learn a model for the accompaniment using non-vocal regions in the input magnitude spectrogram, which allows subtracting the estimated accompaniment from vocal regions. The algorithm was tested in sepa-

ration of both real commercial music and synthesized acoustic material, and produced clearly better results than the reference separation algorithms. The proposed method was also tested in aligning separated vocals with textual lyrics, where it improved slightly the performance of the existing method.

7. References

[1] M. Wu, D. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.

[2] 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*, vol. 43, no. 4, 2004.

[3] 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 *IEEE International Symposium on Multimedia*, San Diego, USA, 2006.

[4] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, 2007.

[5] B. Raj, P. Smaragdis, M. Shashanka, and R. Singh, "Separating a foreground singer from background music," in *International Symposium on Frontiers of Research on Speech and Music*, Mysore, India, 2007.

[6] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single channel source separation and its application to voice / music separation in popular songs," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, 2007.

[7] M. Ryynänen and A. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, 2008, to appear.

[8] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of Neural Information Processing Systems*, Denver, USA, 2000, pp. 556–562.

[9] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the 5th International Symposium on Independent Component Analysis and Blind Signal Separation*, Granada, Spain, September 2004.

[10] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[11] 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 *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Las Vegas, USA, 2008.

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