TOWARDS LIGHT-WEIGHT, REAL-TIME-CAPABLE SINGING VOICE DETECTION

Bernhard Lehner, Reinhard Sonnleitner, Gerhard Widmer
Department of Computational Perception
Johannes Kepler University of Linz

ABSTRACT

We present a study that indicates that singing voice detection – the problem of identifying those parts of a polyphonic audio recording where one or several persons sing(s) – can be realised with substantially fewer (and less expensive) features than used in current state-of-the-art methods.

Essentially, we show that MFCCs alone, if appropriately optimised and used with a suitable classifier, are sufficient to achieve detection results that seem on par with the state of the art – at least as far as this can be ascertained by direct, fair comparisons to existing systems. To make this comparison, we select three relevant publications from the literature where publicly accessible training/test data were used, and where the experimental setup is described in enough detail for us to perform fair comparison experiments.

The result of the experiments is that with our simple, optimised MFCC-based classifier we achieve at least comparable identification results, but with (in some cases much) less computational effort, and without any need for extensive lookahead, thus paving the way to on-line, real-time voice detection applications.

1. INTRODUCTION

Identifying the regions in a song where a singing voice is present does not seem to be a difficult task for humans, regardless of the singer's specific voice characteristics, dynamics of articulation, instrumental background, or even the language. However, the automatic classification of vocals remains difficult, to a considerable degree due to the extreme extent of vocal tone diversity. At the same time, automatic singing voice detection would be extremely useful for many applications such as audio segmentation and indexing, language detection, singer recognition, vocal extraction, query-by-lyrics, real-time tracking and synchronisation, etc.

Consequently, there has been a lot of research recently into this problem. A multitude of diverse audio features have been proposed, along with many (and sometimes complex) detection methods. In extensive experimental work, where we tried to reproduce some of this work and determine what the most useful features are, we were surprised to find that, when we invested a lot of effort into searching for optimal parametrisations of the features, in the end simple MFCCs always turned out to be at least as good as larger and more complex sets of features.

That is the starting point for this paper, in which we will demonstrate that singing voice detection of state-of-the-art quality can be done in a very light-weight way, using only appropriately parametrised MFCCs. We will ascertain this by comparing our very simple method to three selected methods from the literature where publicly accessible training/test data were used, and where the experimental setup is described in enough detail for us to perform fair comparison experiments.

The result of the experiments is that with our simple, optimised MFCC-based classifier we achieve at least comparable identification results, but with (in some cases much) less computational effort, and without any need for extensive lookahead, thus paving the way to on-line, real-time voice detection applications.

The paper is structured as follows: Section 2 gives an overview of previous work on singing voice detection and on publicly available corpora for this task, and motivates the choice of state-of-the-art methods that we choose to compare our method to. Section 3 presents our own simple method and describes how we experimentally optimised the MFCC and classifier parameters. Section 4 compares our method to the other methods selected above, and Section 5 outlines what we consider to be the most important direction for further improvement.

2. PREVIOUS WORK AND AVAILABLE DATA

We start with a brief overview of selected existing methods, and of publicly available data corpora. Finally, we identify the methods we chose to compare our results to, along with the reasons why we chose those methods.

2.1 Problem statement

The signal consists only of music and the problem is to detect the presence of singing voice therein. Hence, no discrimination of normal speech and singing voice is done, in contrast to the speech/music discrimination task done by Chou and Gu in [4].

2.2 Previous Work on Singing Voice Detection

In [21], Rocamora and Herrera compared MFCCs, Perceptually derived LPC (PLPC), Log Frequency Power Coefficients (LFPC) and Harmonic Coefficient (HC). Spectral features often utilised for instruments classification were also combined into one vector, namely Centroid, Roll-off, Flux, Skewness, Kurtosis, and Flatness. Additionally, pitch as the only non-spectral feature, extracted with the monophonic f0-estimator YIN [5] was used. Several empirically motivated post-processing strategies were also implemented. In their setting, MFCCs are the most suitable features, and an SVM the best performing classifier for singing voice detection. Although they tried to combine different descriptors, this did not improve the classification performance of 78.5% accuracy on a test set of 46 manually annotated songs. Since this material is not publicly available, it is not possible to conduct a fair comparison.

In [16], HA-LFPCs (harmonic attenuated LFPC) are proposed for singing voice detection. For the construction of the attenuation filter, the key determination technique from [23] is used. Compared to MFCCs, the LFPCs performed better, with a Multi-Model HMM (taking song structure into account) as classifier (86.7% vs. 81.3% accuracy). The experiments were carried out on 20 unknown popular songs from different artists and time spans, with six songs used for training and the remaining 14 songs for testing. Since this material is not publicly available, it is not possible to conduct a fair comparison.

Li and Wang [14] used a singing voice detection stage for the separation of singing voice from instrumental accompaniment. They used MFCCs, LPCs, PLPs, and the 4-Hz harmonic coefficient as features. A HMM and the Viterbi algorithm [18] were used to classify clips from five rock and five country songs via a 10-fold CV. Again, this musical material is not publicly available.

Regnier and Peeters' method [20] involves thresholds for vibrato and tremolo to detect singing voice. They reached an f-value of 76.8% for singing voice compared to a (more sophisticated) machine learning approach using MFCCs, SFM, and their first and second derivatives for both features and a GMM as classifier, which yielded 77.4% f-measure. The experiments were conducted on the Jamendo corpus [19], with 63 songs as training set, and 32 songs as the test set. Since the results reported are not as good as those reported in [19], which were also obtained on the Jamendo corpus, we will compare our method to the latter.

Hsu and Jang [11] used GMMs as states in a fully connected HMM and the Viterbi algorithm [18] to decode music signals into the three classes *accompaniment, unvoiced* and *voiced*. They achieved an accuracy of 77.95% with a 39-dimensional feature vector containing 12 MFCCs, the log energy, and their first and second derivatives. They used all 1000 clips of the MIR-1K data set [11] for training and evaluating, divided into two subsets of similar sizes (487 versus 513, recorded by disjoint subjects) for a 2-fold CV. Since the precise split of the data set is not publicly available, it is not possible to conduct a fair comparison.

In [12], Hsu et al. used basically the same setting as in [11], except that they used Harmonic/percussive source separation (HPSS) as a preprocessing step. Additionally, their HMM decodes into just vocal and nonvocal sections. The usage of the preprocessed signal showed significant improvement compared to the raw signal, especially in lower SNR levels. For an -5, 0, and 5 dB SNR they reached accuracies of $\sim 80\%$, $\sim 85\%$, and $\sim 89\%$ respectively.

The three methods of Vembu and Baumann [25], Mauch et al. [15] and Ramona et al. [19], to which we will compare our own method, will be described in more detail in 3.4, 4.2, and 4.1 respectively. The reasons why we chose them are explained in Section 2.4.

2.3 Publicly Available Corpora

As outlined above, numerous very different approaches have been taken to the problem of singing voice detection. Unfortunately, due to a lack of commonly used corpora, the results are often not directly comparable. To our knowledge, there are three corpora along with vocal activity annotations publicly available:

1. **RWC Music Database: Popular Music (RWC-MDB-P-2001):** 100 songs released by Goto et al. [8], with singing voice annotations provided by Mauch et al. [15]. Along with the annotations a novel method was introduced which will be described in more detail in Section 4.2.

2. **Jamendo Corpus:** 93 copyright-free songs from the Jamendo music sharing website [9], collected and annotated by Ramona et al. [19]. Also used for singing voice detection by Regnier and Peeters in [20].

3. **MIR-1K Corpus:** 1000 song clips taken from 110 karaoke songs, and released by Hsu and Jang in [11]. The songs were recorded in their lab, and sung by 8 females and 11 males. Also used for singing voice detection by Hsu et al. in [12].

2.4 Algorithms Selected for Comparison

Eventually, we selected three different methods as benchmarks for our proposed method.

2.4.1 Vembu and Baumann [25]

This method is relatively simple and achieved remarkable results on an unknown test set. The method is described in such detail that it can be re-implemented. It will be described in more detail in subsection 3.4. Thus, our re-implementation of it will be used as a "baseline" to compare our simple optimised MFCC-only classifier to (see Section 3.4). We will show that MFCCs alone can achieve better performance, if appropriately parametrised.

2.4.2 Ramona et al. [19]

For this method, a very large and diverse set of features is used. They precisely report (also song-wise) results on a publicly available corpus (Jamendo) with exact information regarding which pieces were used for training and

evaluation. This allows for the most fair and precise comparison. The method will be described in more detail in subsection 4.1.

2.4.3 Mauch et al. [15]

This is one of the most recent publications on this topic. The authors report excellent results on (parts of) the publicly available RWC corpus, with a rather complex procedure. This method will be described in more detail in subsection 4.2.

3. A SIMPLE MFCC-BASED RECOGNISER

In this section, we describe how we designed and optimised our simple singing voice detector using solely MFCC features. The optimisation was done in three phases, and we will show, for each phase, how the improvements compare to our "baseline" algorithm from Vembu and Baumann [25]. This will give a first impression of results achievable with MFCCs alone.

3.1 Classification Setting and Basic Features

Our classification setting is as follows. The units of audio to be classified are frames of 200 ms duration. Thus, we have 5 classifications per second of audio. The actual window over which the features for classifying a frame F are computed, may be larger than that. We will call this the *observation window*. It will always be placed symmetrically around a classification frame F. We use the VOICEBOX toolbox [3] to extract a D-dimensional MFCC feature vector from the observation window. As the standard parameters we used 30 triangle shaped filters and extracted 13 coefficients, without the 0th coefficient.

For the parameter optimisation phase, we used a set of 75 annotated songs by 75 different artists, which come from a different source than the corpora we will use for the comparison experiments below. All songs were unified, i.e. downsampled to 22kHz and converted to mono. Approximately 52% of the frames are annotated as vocal, and the amount of pure singing, i.e. without instrumental accompaniment, is negligible. All optimisation decisions in the following are based on 15-fold cross validation (CV) experiments, where the data set was randomly split into 15 subsets of 5 songs each.

3.2 Classifier and Post-Processing

Among the most popular classifiers for singing voice detection are Gaussian Mixture Models (GMM), Support Vector Machines (SVM) and Multi-layer perceptron neural networks (MLP). Random forests [2] have proven to deliver good results in other contexts, for instance in speech detection in [24] or music detection in [22]. Compared to SVMs and MLPs they perform much faster in both the training and testing phase. Also, there is no need to determine an appropriate kernel function in order to obtain the best performance. Thus, we chose to use a random forest as classifier, and the implementation from WEKA [10].

For post-processing (smoothing of the prediction sequence), a simple median filter with a window length of seven frames (1.4s) was found to give the best trade-off between complexity and accuracy.

3.3 Optimising the MFCC Features

In this subsection we describe the task of the parameter optimisation, which was done in three phases:

3.3.1 Phase I:

The number of coefficients as well as the size of the filterbank was optimised. The best results were achieved with 30 coefficients (including the 0th) and a filterbank with 30 triangular shaped filters.

3.3.2 Phase II:

The length of the observation window was optimised. In all cases, the observation window, when it was larger than the current classification frame, was placed symmetrically around the frame. Best results were achieved with a total window of 800ms around the 200ms center frame. The relatively long observation window has the effect that the contribution of percussive components (which are only active for a short period of time) to the spectrum is diminished. Before the last phase was conducted, the first derivatives (deltas) of the MFCCs were also added to the feature vector.

3.3.3 Phase III:

The parameters of the classifier were optimised. A good trade-off between computational complexity and the results was found to be a random forest with 128 trees, each with five attributes. Additionally, the threshold for the vocal class was raised to 55% to reduce some of the false positives.

3.4 Comparison to Vembu and Baumann

Vembu and Baumann [25] extracted 13 MFCCs, 39 PLPs, and 12 LFPCs, resulting in a feature vector with 64 elements. They achieved the best results with all features combined and using a SVM as classifier (93.47% accuracy on an unknown data set). Additionally, they provide the parameters of the optimised RBF kernel they used ($C = 2^8$, $\sigma = 2^2$). They do not mention any post-processing. Since there is enough information available to implement this method, it will be used as a baseline for further comparison. Although theirs is an online algorithm, it turns out that training and testing is extremely time consuming (about 75 times slower than with our optimised random forest).

Table 1 shows the results of the 15-fold CV on our data set (see Section 3.1), at different stages of the optimisation procedure, along with the results of the VB method (computed on exactly the same data splits). To illustrate the effectiveness of every optimisation stage, we begin with the results achieved with standard MFCCs (see Section 3.1) and a standard random forest classifier (column MFCC). After the first optimisation stage (I), where we use 30 coefficients instead of just 13, there is an improvement in

	MFCC	I	II	PROP	VB
acc [%]	69.14	74.51	78.74	82.36	77.16
recall	0.727	0.783	0.834	0.883	0.819
precision	0.712	0.757	0.788	0.810	0.774
f-measure	0.719	0.770	0.810	0.845	0.796

Table 1. Results of the parameter optimisation compared to the baseline method VB. The columns are as follows: MFCC: "standard" (unoptimised) MFCCs. I: after optimisation stage I (number of MFCCs; filterbank). II: after optimisation stage II (observation window; delta MFCCs). PROP: proposed method after optimisation III (optimised random forest; median filter). VB: Vembu & Baumann trained and tested in the same way. Recall, precision, and f-measure relate to our class of interest, *vocals*.

accuracy of more than 5 percentage points. The second optimisation regarding the observation time (II) yields an improvement of almost 10 points compared to the standard MFCCs, and already better results than the baseline algorithm of Vembu and Baumann [25] (VB). The final proposed method (PROP.) with the optimised random forest classifier and median-filter post-processing, reaches an accuracy 13 percentage points higher than standard MFCCs, and more than 5 points better than the VB method.

4. COMPARISON TO STATE-OF-THE-ART METHODS

In this section our proposed simple method is compared to two other methods. In Section 2.4 we already explained the motivation behind the selection of the algorithms; now we explain them in more detail.

To recapitulate, our proposed method uses just the optimised long-term MFCCs (800ms, 30 coefficients incl. the 0th, and a filterbank with 30 triangular shaped filters) along with their first derivatives. There is no pre-processing involved, and a simple median filter over seven frames (1.4sec) is used to smooth out the predictions of a random forest classifier (128 trees with five features each).

4.1 Ramona et al.

Ramona et al. [19] use an SVM classifier with a combination of the most diverse set of features compared to the other methods discussed in this paper. These include MFCCs, LPCs, ZCR, sharpness, spread, f0 and aperiodicity measure extracted with the monophonic YIN library [5]. Furthermore, short-scale frames contain spectral descriptors like centroid, width, asymmetry, slope, decreasing, flux, and similar temporal statistical moments. Additionally, long-scale frames contain features that do not represent an instantaneous characteristic. Those include (again) the ZCR, tremolo and granularity for the frequencies 4-8Hz and 10-40Hz, and some temporal statistical moments. Those features add up to a vector with 116 components.

	Ramona et al.		PROP		VB	
	Acc%	F%	Acc%	F%	Acc%	F%
03 - Say me Good Bye	80.1	85.8	91.4	83.6	90.6	82.7
03 - School	84.3	87.3	84.8	86.6	71.5	77.8
03 - Si Dieu	76.4	80.7	87.2	89.4	76.2	66.7
03 - Une charogne	85.3	91.7	89.8	93.5	78.8	85.9
03 - castaway	79.0	87.3	71.3	80.5	73.0	80.0
04 - Believe	80.0	88.5	94.1	95.6	83.0	87.2
04 - Healing Luna	85.5	81.6	87.8	84.4	72.7	70.8
04 - Inside	83.3	68.2	79.4	66.0	75.3	58.0
04 - You are	87.0	91.9	87.9	90.6	74.4	77.7
05 - 05 LIrlandaise	57.7	64.2	65.0	68.6	61.7	60.5
05 - 16 ans	91.5	84.8	87.3	79.8	70.8	60.3
05 - 2003-Circons[...]	87.6	88.2	75.5	77.7	79.8	79.6
05 - A Poings Fermes	93.7	92.2	89.7	83.0	86.9	81.2
05 - Crepuscule	85.2	88.8	80.1	83.6	76.8	80.0
05 - Dance	77.0	83.2	84.1	88.7	75.7	82.2
05 - Elles disent	71.8	78.7	84.4	87.4	69.6	77.0
ALL	82.2	84.3	84.8	84.6	77.4	76.9

Table 2. Results of the proposed method on the Jamendo corpus, compared to those of Ramona et al. in [19] and Vembu and Baumann's method trained and tested in the same way.

Afterwards, the dimensionality is reduced to d=40 with the IRMFSP algorithm [17], leaving only the most discriminating features. A silence detection is applied as a pre-processing step. Finally, a HMM and the Viterbi algorithm are used for post-processing the SVM output, and instrumental segments shorter than 0.5s are discarded.

The authors report 82.2% accuracy on a precisely described split of the Jamendo corpus, with a training set consisting of 63 given songs, and validation and test sets of 16 songs each. Thus, a fair comparison of the results is possible.

In Table 2, the results of Ramona et al. are compared to those of the proposed method. All in all, better results regarding both accuracy and f-measure are achieved with the proposed method (82.2% vs. 84.8% accuracy). Nevertheless, there are some songs that get better classified with the method of Ramona et al.; the biggest difference is with the song *05 - 2003-Circons[...]* (87.6% vs. 75.5% accuracy). It would be interesting to have a feature with which we could determine the better suited method for a specific song, or even a shorter segment.

4.2 Mauch et al.

Mauch et al. [15] utilise four features in total, among them MFCCs. Additionally, they use Goto's polyphonic fundamental frequency(f0)-estimator PreFEst [7] to isolate the predominant melody. They propose three novel features which are based on it:
Pitch fluctuation, which is basically the frame-wise standard deviation of intra-semitone f0 differences. First, the estimated f0 is mapped to pitch space. Afterwards, these estimations are shifted based on a song-wide inferred tuning. As a last step, the frequency differences are calculated, and the frame-wise Hamming-weighted standard deviation of those differences yields the pitch fluctuation. Since a song-wide lookahead is necessary to infer its tuning, this method is not an online algorithm. Pitch fluctuation is

found to be the most salient feature for singing voice detection.

In addition to the MFCCs of the signal as it is, the authors also introduce *MFCCs of the re-synthesised predominant voice* to capture its timbre. The re-synthesis employs sinusoidal modelling based on the predominant melody as well as the estimated amplitudes of its harmonics as described in [6].

The *normalised amplitude of harmonic partials* is also extracted from the predominant voice, and is considered to add information on another dimension of timbre which is not provided by MFCCs. It is a vector-shaped feature (d=12), and calculated by normalising the estimated harmonic amplitudes according to the Euclidean norm.

A SVM-HMM [1, 13] is used as classifier. Additionally, segments shorter than 0.5s are merged with the preceding regions.

The best result (87.2% accuracy) was achieved with all four features combined, employing a 5-fold CV on a 102 song data set that is composed of 90 songs from the RWC music database [8] (exactly which 90 of the 100 is unknown to us and could, unfortunately, not be found out), and 12 additional (also unknown) songs. Since we had only access to the 100 song RWC music database, our results are only comparable to a certain extent. Nevertheless, to allow for the best comparison possible, we matched their decision frequency of one feature instance every 100ms and utilised a 5-fold CV.

In Table 3, the results of Mauch et al. are compared to those of the proposed method. To illustrate the difference of the data set used by Mauch et al. to the original RWC data set, we also give the results achieved with standard MFCCs (see Section 3.1). Additionally, to reveal the amount of vocals, we give the results of the mode, i.e. the proportion of the majority class (which is *vocals*). As can be seen, there is a difference of 5 percentage points regarding the vocal content, which indicates a limited comparability.

All in all, our proposed method performs not much worse than the method of Mauch et al. (85.9% vs. 87.2% accuracy). This difference virtually disappears when we apply a more complex post-processing strategy involving a HMM and the Viterbi algorithm (row PROP+).

Mauch	**accuracy**	**precision**	**recall**	**f-measure**
MODE	0.654	0.654	1.000	0.791
MFCC	0.738	0.739	0.926	0.822
FMRH	0.872	0.887	0.921	0.904
Proposed	**accuracy**	**precision**	**recall**	**f-measure**
MODE	0.604	0.604	1.000	0.753
MFCC	0.718	0.764	0.771	0.767
VB	0.813	0.827	0.808	0.818
PROP	0.859	0.858	0.918	0.887
PROP+	0.868	0.879	0.906	0.892

Table 3. The results of our proposed method compared to the methods of Mauch et al. and Vembu and Baumann. Along with the methods the class distribution in the respective test set is given (row MODE – the overall proportion of vocals), as well as the results achieved with the standard MFCCs, and Vembu and Baumann's Method (row VB). Clearly, even though the majority of the data we used is the same as used by Mauch et al., there are differences regarding the content of vocals, which makes a fair comparison unfeasible. The results PROP+ are achieved with a post-processing involving the Viterbi algorithm.

5. CONCLUSION AND FUTURE WORK

This paper has proposed an extremely simple method to detect the presence of singing voice in mixed audio signals. By comparing the results to those of three well selected algorithms, we could show that regarding the features, appropriately parametrised MFCCs along with their first derivatives are sufficient to achieve results as good as those of sometimes much more complicated state-of-the-art systems. Our method is simple, fast, and requires no look-ahead, making it a good candidate for on-line, real-time singing voice detection applications.

Our main goal for further improvement is *precision*, that is, a reduction of the number of *false positives*. A detailed inspection of the results of our classifier has shown that instruments mistaken for vocals have the biggest negative impact on the results. This is especially true for string instruments like electric guitars, which can mimic the temporal as well as the timbral characteristics of vocals. Certain effects commonly used to enhance the sound or extend the expressiveness of guitars like chorus, flanger, and wah-wah are responsible for this. Unfortunately, experiments reported in [21] indicate that features often utilised for speech/music discrimination like harmonic coefficient [4] are not able to distinguish between highly harmonic instruments and vocals. Therefore, it would be beneficial to develop a method that is less sensitive to differences between singers' specific voice characteristics, while maintaining good discriminative properties for instruments that resemble vocals.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Fund FWF under grants TRP307-N23 and Z159 (Wittgenstein Award).

7. REFERENCES

[1] Y. Altun, I. Tsochantaridis, T. Hofmann, et al. "Hidden Markov support vector machines". In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, volume 20, 2003.

[2] L. Breiman. "Random forests". *Machine learning*, 45(1):5–32, 2001.

[3] M. Brookes. "Voicebox: Speech Processing Toolbox for Matlab". Website, 1999. Available online at `http://www.ee.ic.ac.uk/hp/staff/dmb/`

voicebox/voicebox.html; visited on March 4th 2013.

[4] W. Chou and L. Gu. "Robust singing detection in speech/music discriminator design". In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001*, volume 2, pages 865–868. IEEE, 2001.

[5] A. De Cheveigné and H. Kawahara. "YIN, a fundamental frequency estimator for speech and music". *The Journal of the Acoustical Society of America*, 111:1917–1930, 2002.

[6] 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 *Proceedings of the Eighth IEEE International Symposium on Multimedia, ISM 2006*, pages 257–264. IEEE, 2006.

[7] 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*, 43(4):311–329, 2004.

[8] 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)*, volume 2, pages 287–288, 2002.

[9] P. Grard, L. Kratz, and S. Zimmer. "Jamendo, open your ears". Website, 2005. Available online at http://www.jamendo.com; visited on March 18th 2013.

[10] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. "The WEKA data mining software: an update". *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[11] C-L. Hsu and J-S. R. Jang. "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset". *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2010.

[12] C-L. Hsu, DL. Wang, J-S. R. Jang, and K. Hu. "A Tandem Algorithm for Singing Pitch Extraction and Voice Separation From Music Accompaniment". *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1482–1491, 2012.

[13] T. Joachims, T. Finley, and C. J. Yu. "Cutting-plane training of structural SVMs". *Machine Learning*, 77(1):27–59, 2009.

[14] Y. Li and DL. Wang. "Separation of singing voice from music accompaniment for monaural recordings". *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1475–1487, 2007.

[15] M. Mauch, H. Fujihara, K. Yoshii, and M. Goto. "Timbre and Melody Features for the Recognition of Vocal Activity and Instrumental Solos in Polyphonic Music". In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 233–238, 2011.

[16] T. L. Nwe, A. Shenoy, and Y. Wang. "Singing voice detection in popular music". In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 324–327. ACM, 2004.

[17] G. Peeters. "Automatic Classification of Large Musical Instrument Databases Using Hierarchical Classifiers with Inertia Ratio Maximization". In *115th AES Convention*, 2003.

[18] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". *Proceedings of the IEEE*, 77(2):257–286, 1989.

[19] M. Ramona, G. Richard, and B. David. "Vocal detection in music with support vector machines". In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008*, pages 1885–1888. IEEE, 2008.

[20] L. Regnier and G. Peeters. "Singing voice detection in music tracks using direct voice vibrato detection". In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, pages 1685–1688. IEEE, 2009.

[21] M. Rocamora and P. Herrera. "Comparing audio descriptors for singing voice detection in music audio files". In *Brazilian Symposium on Computer Music, 11th. San Pablo, Brazil*, volume 26, page 27, 2007.

[22] K. Seyerlehner, T. Pohle, M. Schedl, and G. Widmer. "Automatic music detection in television productions". In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx'07)*, 2007.

[23] A. Shenoy, R. Mohapatra, and Y. Wang. "Key determination of acoustic musical signals". In *Proceedings of the 2004 IEEE Conference on Multimedia and Expo, ICME 2004*, volume 3, pages 1771–1774. IEEE, 2004.

[24] R. Sonnleitner, B. Niedermayer, G. Widmer, and J. Schlüter. "A Simple And Effective Spectral Feature For Speech Detection In Mixed Audio Signals". In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx'12)*, 2012.

[25] S. Vembu and S. Baumann. "Separation of vocals from polyphonic audio recordings". In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, volume 5, pages 337–344, 2005.