# Detecting key features in popular music: case study – singing voice detection

Rui Nóbrega, Sofia Cavaco

CITI, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
{rui.nobrega,sc}@di.fct.unl.pt

**Abstract.** Detecting distinct features in modern pop music is an important problem that can have significant applications in areas such as multimedia entertainment. They can be used, for example, to give a visually coherent representation of the sound. We propose to integrate a singing voice detector with a multimedia, multi-touch game where the user has to perform simple tasks at certain key points in the music. While the ultimate goal is to automatically create visual content in response to features extracted from the music, here we give special focus to the detection of voice segments in music songs. The solution presented extracts the Mel-Frequency Cepstral Coefficients of the sound and uses a Hidden Markov Model to infer if the sound has voice. The classification rate obtained is high when compared to other singing voice detectors that use Mel-Frequency Cepstral Coefficients.

**Keywords:** Singing voice detection, features detection, music game, visual sound synchronization, MFCC, Hidden Markov Model

## 1 Introduction

The sound component plays a vital part in today's interactive games and applications. Entertainment applications with good graphics lose their appeal when the sound is not present or is poorly integrated. Here we propose a singing voice detector integrated with an interactive game that explores a multi-touch interface [7]. The application expects the user to respond with multi-touch gestures to certain events in its background music, such as a beat, the start of a voice, a guitar, or other instrument. To illustrate what sequences of movements can be made, the application has some *training* examples that should appear before the events. As a first step to automate the synchronization between the examples and the music, we have implemented the singing voice detector presented here.

Our singing voice detector uses speech recognition strategies to detect voice segments in small music samples. It extracts Mel-Frequency Cepstral Coefficients (MFCCs) from music segments and uses them to train a Hidden Markov Model (HMM). This model is then used to classify consecutive small blocks of the sound. To evaluate the solution we compared the detected blocks with previously manually classified songs.

## 2 Related Work

There has been extensive research in features detection and audio classification. Breebaart and McKinney summarize several features sets for audio classification in [1]. Harte *et al.* present a method to detect harmonic change in musical audio based on pitch differences between two consecutive sound frames, and which can be useful in singing voice detection [2].

Specific voice detection in music is not a widely studied subject, but most of its ideas come from speech recognition topics which is a well established research area. Khine *et al.* define and analyze four acoustic features: vibrato, harmonicity, timbre and cepstral coefficients computation such as MFCCs [3]. The first three features are used essentially as a cue to detect voice segments. Then, the MFCCs of voice classified songs are used to train a HMM. The final model detects vocal segments in songs which are then inserted back into the HMM. This work presents a generic model and some ideas but lacks a complete solution description.

Using a very similar strategy, Nwe *et al.* present several experiments in singing voice detection but with different cepstral coefficients [8]. While Khine *et al.* used Octave Frequency Cepstral Coefficients (OFCC) and MFCCs (with no major differences in the two methods, 80% success, but slightly better results when combined, 83% success) [3], Nwe *et al.* tested four types of coefficients and concluded that the best results (86.7%) were given by the Harmonic Attenuated Log Frequency Power Coefficients (HA-LFPC). These coefficients are obtained by using a triangular bandpass filter on the spectrogram which reduces the amplitude of harmonic sounds. This way non-vocal sounds will have much less energy than vocal sounds. It must be stated that the authors do not specify what type of musical instruments are being used in their tests, probably most of them have harmonic sounds, thus making this method have a larger success rate. In the same tests MFCCs obtained an average of 81.3% of success.

Li and Wang present a complete solution to separate singing voice from music, which is actually a harder problem than just voice detection [4]. Their system has three steps: first the voice segments are detected, then the predominant pitch is detected and finally the voice is separated from the rest of the music. The singing voice detection has several stages. First of all a spectral change detector selects several segments of audio where the energy has significant spectral changes between frames. After the input is portioned, it is classified as vocal or non-vocal by a HMM likelihood function. This method appears to be very similar to the ones presented in [3,8] using MFCCs to train HMMs but has additional details about implementation issues. The success rate of this method, which uses one HMM for vocal sounds and another for non-vocal sounds, is around 79%. Note that the evaluation process used the same samples that served as input for the training of the HMM, which might have produced biased results. (Papers [3,8] do not state the origin of the samples.)

Hidden Markov Model has many applications such as in speech, handwriting and musical score recognition. In this work it will be used for singing voice recognition. Russel and Norvig in their book [10] dedicate an entire chapter to statistical methods and probabilistic reasoning over time. Another important effort is the work done by Rabiner in [9] where a complete tutorial for HMMs is presented and some applications of them to speech recognition are discussed.

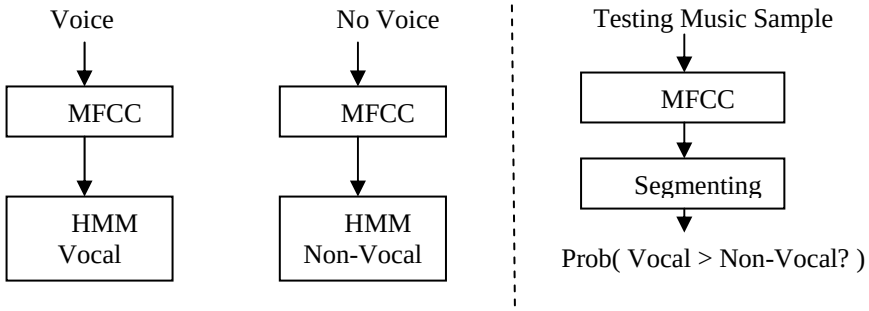


**Fig. 1.** System solution diagram. It initially trains two HMMs and uses them to classify music samples.

### 2.1 Mel-Frequency Cepstral Coefficients

MFCCs are coefficients that try to capture the perceptual information of sound. They have many applications in sound retrieval, in genre classification, audio similarity measure or speech recognition. The MFCCs are based on the Mel-scale, which is a perceptual scale of pitches. This scale is based on the fact that the correlation between human perceived pitch distance between two sounds is not linear in frequency. It is easier for a person to distinguish two low frequency sounds, such as 300Hz and 400Hz, than it is to distinguish between two high frequency sounds, such as 6000Hz and 6100Hz. For this reason the Mel-scale is a logarithmic function.

In order to extract the MFCCs [11] of a sound segment several steps have to be done. Sigurdsson *et al.* explain, test and compare several implementations of MFCCs. The most common extraction methods follow these steps: division of the signal into several frames, computation of the Fast Fourier Transform (FFT), filtering the FFT results by a Mel filter bank, take the log of the powers at each Mel frequency and find the Discrete Cosine Transform (DCT). The MFCCs are the amplitudes of the resulting spectrum.

The voice detector's Mel Filter bank, which is from the Auditory Toolbox [12], is constructed using 13 linearly-spaced filters (133.33Hz between center frequencies) followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency). The amplitude of the FFT frames is then combined with the Mel filter bank. The final result is 13 coefficients for each frame. These are the values used in the HMM.

## 3 Solution

Following ideas presented in [3,4,8], our singing voice detector uses two HMMs: one for voice and another for non-voice music segments. The overall solution is illustrated in figure 1. Two HMMs are trained with the MFCCs of two sets of training sounds (one HMM is trained with data corresponding to vocal samples and the other with non-vocal samples). The music segmentation (into vocal and non-vocal segments) is

done comparing the log-likelihood probability of the two HMMs. The output of the application is the set of ranges where voice was detected.

The application was built in Matlab, using the MFCCs from the Auditory Toolbox [12] and the HMM toolbox [6]. To train it, a set of sounds was gathered, half of them containing voices. Several two second samples were taken from different songs and from different artists. Half of the samples had voice. The samples had a quality of 44100Hz, 16 bits, Mono. The entire algorithm works only on a single channel but it would be trivial to replicate it for stereo sound. The initial research was done in female voices, in pop-rock songs using short music segments with only one voice at a time.

Once the samples are fed into the system, they are divided into small frames and converted into MFCC vectors. The samples are divided into 100 frames per second, and a vector of 13 MFCCs is obtained for each frame, thus resulting in a matrix $D$ of 13 (coefficients) by 200 (observations), for each two second sample.

The two HMMs are trained with the MFCCs. The HMMs were defined using as input $D$. Since sound is a continuous signal the possible values of the coefficients are not discrete symbols of a finite alphabet. For that reason we use a Gaussian Mixture model with $M$ mixtures to represent $Q$ possible states. $M$ and $Q$ are left open as parameters. Finally the HMMs are generated using an iterative process which tries to approximate the model to best describe the training sound set (about 1000 iterations were used here).

Once the two HMMs are generated, they can be used to segment the game's background music, according to its voice segments. In summary, the system reads the music and converts it into $n$ MFCC vectors. Then it takes the resulting 13 by $n$ matrix and divides it into segments of a given size (we used size 40 in all tests). Finally it uses this data to compute the likelihood probability of it belonging to the vocal or to the non-vocal HMM class. The segments will be classified as belonging to the class with higher probability.

## 4 Evaluation and Results

To evaluate the method several samples with voice and instruments, and samples with no voice (only instruments) were manually classified. To do this in real time, a small application was built where a person would listen to a song and would push a button whenever the singer started and ended singing. This is a very fast method to obtain classified data although it is prone to some errors and possible delays associated with the person's reaction time. Using the classified songs it was possible to evaluate the results from the proposed algorithm (figure 1). The example in figure 2 shows that the automatic method detects almost all the voice segments. There are some false positives in some high energy segments (ex: a beat with eco) and some false negatives when the voice volume is very low.

In order to measure the algorithm's performance several tests where made with different values for the HMMs' parameters and training data. First of all the HMMs were trained and tested using samples from the same band of the music that it would be tested. Then they were trained using samples from more bands. Finally we tried

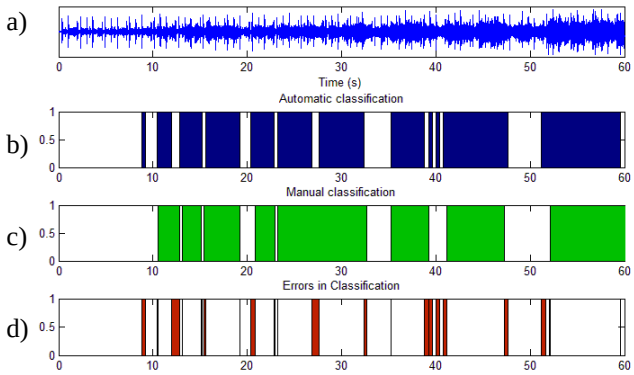


**Fig. 2.** Singing voice classification. (a) Music´s waveform. The dark areas represent the segments that are classified as vocal by the algorithm (b) and manually (c). The last graph (d) shows the differences between the other two. In this 60 second example the success rate was 90%.

with different values for the number of Gaussian mixtures, *M*, and the number of states, *Q*. The success rate of each approach can be seen in table 1. Note that the training set was composed of small samples of songs and the testing set was composed of full songs independent from the samples.

The results had a variation of around ±5% due to different seeds from the training algorithm. The tests had a slightly better result when the training set is only composed of samples from the songs being tested (83.1%). The other training set had samples from the song being tested along with samples from different music bands. For this reason the HMM's transition probabilities may become more diluted, therefore resulting in worst results. The best results are obtained using fewer states *Q*. There are not large differences between the results but it is safe to say that using between 1 and 4 states with 2 to 10 Gaussian Mixtures is a good parameter solution.

**Table 1.** Success rate with different parameters. *M* represents the number of Gaussian mixtures and *Q* represents the number of states.

| % | | ***Q*** | | | |
|---|---|---|---|---|---|
| ***Train Samples*** | ***M*** | 1 | 4 | 8 | 16 |
| Same band from test | 2 | 82.4 | 82.3 | 81.2 | 80.8 |
| | 10 | 83.1 | 82.8 | 77 | 76.6 |
| All bands | 2 | 80.3 | 80.4 | 81.4 | 76.4 |
| | 10 | 81 | 80.1 | 78.1 | 79.4 |

## 5 Conclusions

Here, we proposed an integration of a singing voice detector with a multi-touch interactive system that shows that it is possible to build multimedia applications that react to a given music. We have shown that a solution based on HMM using MFCCs

as features is a valid answer to detect singing voice in music. The results show a slightly higher degree of success when compared with previous detectors that use MFCCs (we obtained 83.1% against around 80% [3,4,8]) although this is difficult to evaluate without using a common data set. The system proposed uses the output of the singing voice detector to identify key points in the music (like the start of a singing voice) and build *training* examples of sequences of movements (to *train* the user) that are in harmony with the music. Other interesting key points can be identified by other features from the sound, such as rhythm, instruments playing, loudness or pitch and see how they change through time.

An idea to improve the voice detector is to feed the results back into the model when the degree of confidence in the system starts getting higher (instead of relying only on manual classification). Finally, the system can also be improved by crossing the segments identified by the HMMs with those identified by detectors based on energy, pitch or harmonic frequency.

**Acknowledgments** The authors would like to thank to everyone at the Interactive Multimedia Group (CITI, Universidade Nova de Lisboa) for all the support and input.

## References

1. Breebaart, J., McKinney, M.: Features for audio classification. In Proc. SOIA2002, 2002.
2. Harte, C., Sandler, M., and Gasser, M.: Detecting harmonic change in musical audio. In Proc. AMCMM '06, 2006
3. Khine, S.Z.K., Tin Lay New, Haizhou Li,: Singing voice detection in pop songs using co-training algorithm, . In Proc. of ICASSP 2008, 2008
4. Li, Y., Wang, D.: Separation of Singing Voice From Music Accompaniment for Monaural Recordings, In Audio, Speech, and Language Processing, IEEE Transactions on , 2007
5. Min Xu, Maddage, N.C.: Changsheng Xu; Kankanhalli, M.; Qi Tian, Creating audio keywords for event detection in soccer video,In Multimedia and Expo. ICME '03, 2003
6. Murphy, K., HMM Toolbox for Matlab, 2009 http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
7. Nóbrega, R., Sabino, A., Rodrigues, A., and Correia, N.: Flood Emergency Interaction and Visualization System. In Proc. of,VISUAL'08,2008
8. Nwe, T. L., Shenoy, A., Wang, Y.: Singing voice detection in popular music. In Proc. MULTIMEDIA '04, 2004
9. Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition. In Readings in Speech Recognition, A. Waibel and K. Lee, Eds. Morgan Kaufmann Publishers, San Francisco, 1990
10. Russel, S., Norvig, P., Artificial Inteligence: A Modern Approach, 2nd Edition, Prentice Hall, Cap15, International Edition, 2003
11. Sigurdsson, S., Petersen,K.B., Lehn-Schiøler, T.: Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music, In Proc. ISMIR'06, 2006
12. Slaney, M., Auditory Toolbox, 2009 http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/