System and Method for Automatic Singer Identification

Tong Zhang
Imaging Systems Laboratory
HP Laboratories Palo Alto

HPL-2003-8
January 15th , 2003*

E-mail: tong_zhang@hp.com

singer identification, singing voice detection, music analysis, music classification, music retrieval, audio browsing, music database management

The singer's information is essential in organizing, browsing and retrieving music collections. In this technical report, a system for automatic singer identification is developed which recognizes the singer of a song by analyzing the music signal. Meanwhile, songs which are similar in terms of singer's voice are clustered. The proposed scheme follows the framework of common speaker identification systems, but special efforts are made to distinguish the singing voice from instrumental sounds in a song. A statistical model is trained for each singer's voice with typical song(s) of the singer. Then, for a song to be identified, the starting point of singing voice is detected and a portion of the song is excerpted from that point. Audio features are extracted and matched with singers' voice models in the database. The song is assigned to the model having the best match. Promising results are obtained on a small set of samples, and accuracy rates of around 80% are achieved.

* Internal Accession Date Only Approved for External Publication
To be published in the IEEE International Conference on Multimedia and Expo, 6-9 July 2003, Baltimore, MD

System and Method for Automatic Singer Identification

Tong Zhang

1. Introduction

With digital music becoming more and more popular (such as music CDs and MP3 music downloadable from the internet), music databases, both professional and personal, are growing rapidly. Technologies are demanded for efficient categorization and retrieval of these music collections, so that consumers can be provided with powerful functions for browsing and searching musical content. Among such technologies, is the automatic singer identification of a song, i.e. to recognize the singer of a song by analyzing audio features of the music signal. With this capability provided in a music system, the user can easily get to know the singer's information of an arbitrary song, or retrieve all songs performed by a particular singer in a distributed music database. Furthermore, this technology may be used to cluster songs of similar voices of singers in a music collection, or search for songs which are similar to a query song in terms of the singer's voice.

Currently, singer's information is manually embedded into music files by professional recorders. However, such information is often lacking, or inconsistent in music pieces downloaded from music-exchanging websites (like Napster), or music clips grasped from digital music channels. Moreover, while text information such as the singer's name may be included in a music file, it is impossible to embed audio features such as the timbre of the singer's voice. Thus, it is not possible to cluster songs which are similar in the singer's voice. In general, the technique of automatic singer identification will add significant functionalities to a digital music management system.

There are prior arts on the automatic speaker identification which identifies the speaker of a given speech segment. While the basic idea of the singer identification is also to recognize the voice of a human being, there are significant differences between a singing signal and a speech signal in several aspects. First of all, the singing voice is mixed with musical instrumental sounds in a song, which makes it much more complicated to extract features of the voice. Furthermore, the time-frequency features of a singing voice are quite different from those in a speaking voice.

Up to now, there has been no solution proposed for solving the singer identification problem. Nevertheless, human beings can recognize the voice of a familiar singer, and distinguish similar singing voices by listening to only a small portion of the song. Therefore, we believe that by extracting and analyzing audio features properly, an automatic system should be able to achieve certain degree of singer identification as well.

The rest of the report is organized as follows. An overview of the proposed scheme for singer identification is presented in Section 2. Details about the training part and the identification part of the system are described in Sections 3 and 4, respectively. Experimental results are shown in Section 5. Finally, concluding remarks and future research plans are given in Section 6.

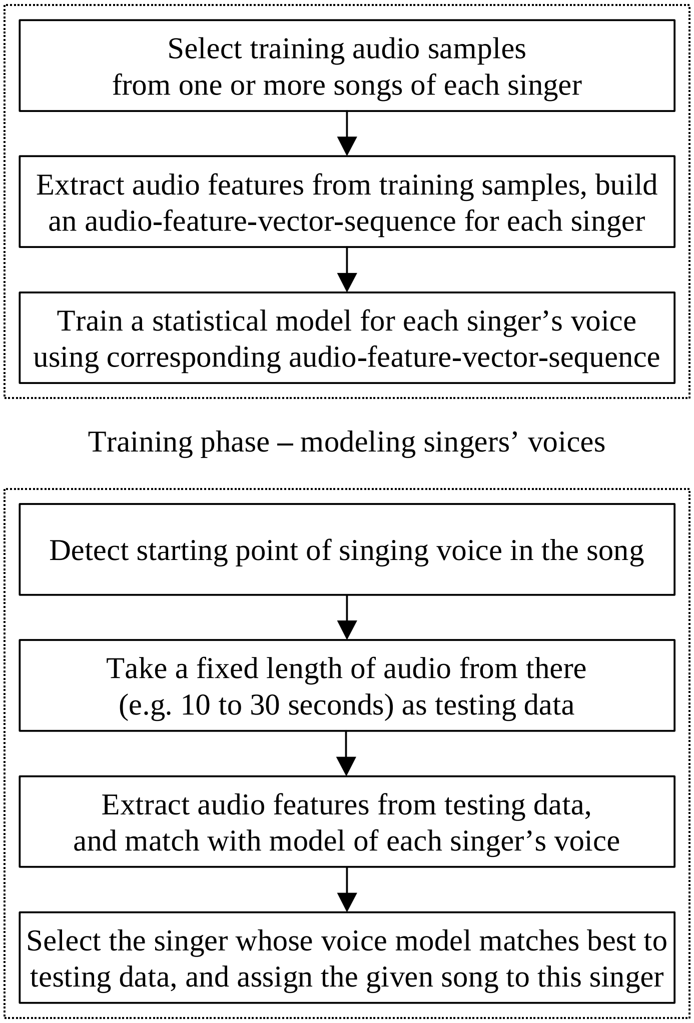
2. Proposed Scheme

The proposed scheme for automatic singer identification contains two phases, as illustrated in Fig.1. One is the training phase, the other is the working phase. In the training phase, training audio samples are selected from one or more typical songs of each singer, and audio features of these samples are computed. These audio features form a feature-vector-sequence which is used to build a statistical model of the singer's voice.

Then, in the working phase, for a song to be identified, the starting point of the singing voice in the song is first detected, and a fixed length of testing data is taken from that point. The testing data are divided into testing samples, and audio features are computed of these samples. The audio features are matched with existing singers' models in the database. If a singer's voice is dominant in one testing sample, its audio features will match well with that singer's voice model, and will have a vote to that model. Otherwise, if a testing sample consists mainly of instrumental sounds, its features may not match well with any of the models, and will not vote to any of the models. In the end, the model which has the most votes will be elected, and the song will be assigned to the corresponding singer.

To cluster songs in a music collection which are of the same singer, or of singers with similar voices, a number of models are trained for different types of singers' voices in the database (one model for each singer, or one model for several singers with similar voices). Then, songs in the music collection are matched with these models and assigned to corresponding clusters. A new cluster and its model are built when a song does not match well with any of existing models.

To retrieve songs that are similar to a query song in terms of singer’s voice, a model is first built for the singer’s voice in the query song. Next, testing samples are extracted from songs in a database, and matched with the voice model. The songs may then be ranked according to the likelihood values, and those songs which sit top at the ranking list are considered to be most similar to the query song in singer’s voice.


Select training audio samples from one or more songs of each singer
Extract audio features from training samples, build an audio-feature-vector-sequence for each singer
Train a statistical model for each singer’s voice using corresponding audio-feature-vector-sequence
Detect starting point of singing voice in the song
Take a fixed length of audio from there (e.g. 10 to 30 seconds) as testing data
Extract audio features from testing data, and match with model of each singer’s voice
Select the singer whose voice model matches best to testing data, and assign the given song to this singer

Training phase – modeling singers’ voices

Working phase – identifying singer of a song

Figure 1: Proposed scheme for singer identification.

3. The Training Process

3.1. Selecting Training Samples

To build a model for a singer's voice, one or more typical songs of this singer are selected. Preludes (i.e. instrumental music played at the beginning of a song) and interludes (instrumental music played between paragraphs and verses of the lyrics) of the song(s) which consist mainly of instrumental sound are manually distinguished and discarded, so that the song(s) are condensed, containing only segments in which the singing voice is dominant. This step is done manually because there is no reliable technique available at present that can automatically detect the singing voice in a song. Next, the condensed song(s) are divided into audio frames. Each frame is 16ms long, with neighboring frames overlapped partially with each other. Each audio frame is then considered as one training sample.

3.2. Extracting Audio Features

Audio features are computed of each training sample to represent characteristics of the singer's voice. There are a number of varieties in choosing and computing audio features for this purpose. For example, MFCC (mel-frequency cepstral coefficients) are the kind of audio feature commonly used in speaker identification and speech recognition systems. By choosing a proper order of the MFCC feature vector, the characteristics of a human voice can be effectively revealed. LPC (linear predicative coding coefficients) is an audio feature indicating characteristics of harmonic components of the audio signal. Since over 90% of the singing signal is harmonic (much more than the case of a speech signal), LPC is also a good choice for representing the features of a singer's voice.

In this work, we compute the LPC cepstral coefficients of each audio frame. The audio signal is first pre-emphasized with a first-order FIR filter. Then, a Hamming window is applied to minimize the signal discontinuities at the borders of each frame. The next processing step is the LPC analysis using the auto-correlation method of order 12. Finally, the LPC parameters are converted to cepstral coefficients using a recursion formula [1], which form a feature-vector of the audio frame. The cepstral coefficients derived from LPC analysis proved to be more robust to noises than the FFT-derived cepstral coefficients, thus are more appropriate to be used with the singing signal which is mixed with instrumental sounds. Feature vectors of all training frames in a singer’s sample data are aligned together to form a training feature-vector-sequence.

3.3.Building Statistical Model of a Singer’s Voice

The training feature-vector-sequence is used to train a statistical model of the singer’s voice. In this work, we choose to use the Gaussian mixture model (GMM). The Gaussian mixture density is a weighted sum of *M* component densities, as follows:

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \qquad (3.1)$$

where $\vec{x}$ is a *D*-dimensional random vector, $b_i(\vec{x})$, $i = 1, \ldots, M$, are the component densities, and p_i, $i = 1, \ldots, M$, are the mixture weights. Each component density is a Gaussian function:

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{ -\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\} \qquad (3.2)$$

with mean vector $\vec{\mu}_i$ and covariance matrix Σ_i. The mixture weights have to satisfy the constraint $\sum_{i=1}^{M} p_i = 1$. A GMM model is built for each singer. We choose the number of mixtures to be 5. There are standard procedures for training a GMM with a feature vector sequence [2]. Resulted parameters of the Gaussian mixture model (including values of the mean vectors, covariance matrixes and weights) from the training procedures are then used to represent characteristics of a singer’s voice.

4. The Identification Process

4.1.Searching for Starting Point of the Singing Voice

To identify the singer of a song, instead of processing the entire song, one portion of the song is excerpted and analyzed. The reason is that on one hand, to process the whole song takes more computation time, especially for songs of longer lengths; on the other hand, prelude, interludes and other segments in the song during which instrumental music is dominant, will negatively affect the identification result. In this work, the starting point of the singing voice in the song is first detected. Then, a fixed length (e.g. between 10 to 30 seconds) of audio signal is taken in the song from the starting point of the singing voice, and used as the testing data. The assumption behind it is that this period of signal in a song normally contains mainly the singer's voice. Moreover, human beings can usually recognize the singer of a song by listening to the first couple of sentences of the song. To detect the starting point of singing voice in a song, four kinds of audio features are used together as follows.

- *The Energy Function*: it represents the amplitude variation over the time of the audio signal [1]. The start of singing voice is normally reflected as a sudden rise in the energy level of the audio signal. A typical example is illustrated in Fig.2 where the short-time energy function of a song is displayed, and the arrow indicates the starting point of the singing voice. Also, in some songs, the appearance of low level local energy minimums after a relatively long period of continuously high energy values may indicate the start of the singing voice.

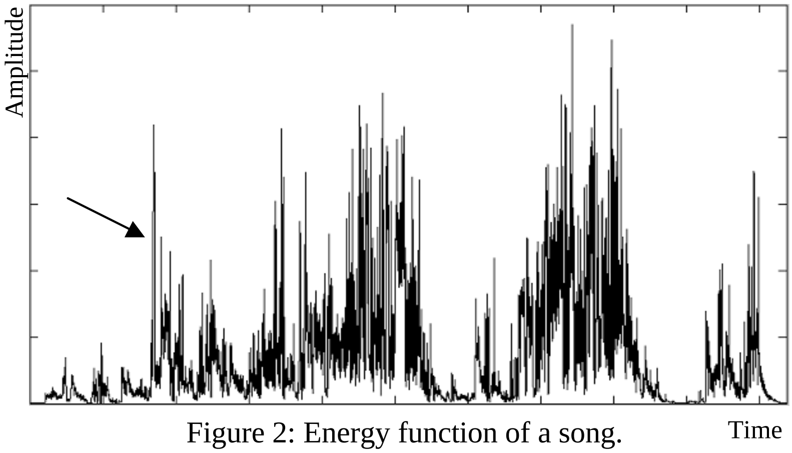


Figure 2: Energy function of a song.

- *The Average Zero-Crossing Rate (ZCR)*: it is a simple measure of the frequency content of the audio signal [1]. While ZCR values of instrumental music are normally within a relatively small range, the singing voice is often indicated by high amplitude ZCR peaks resulted from pronunciations of consonants. An illustration is shown in Fig.3, where the short-time average zero-crossing rates of a song are plotted, and the arrow denotes the start of the singing signal.

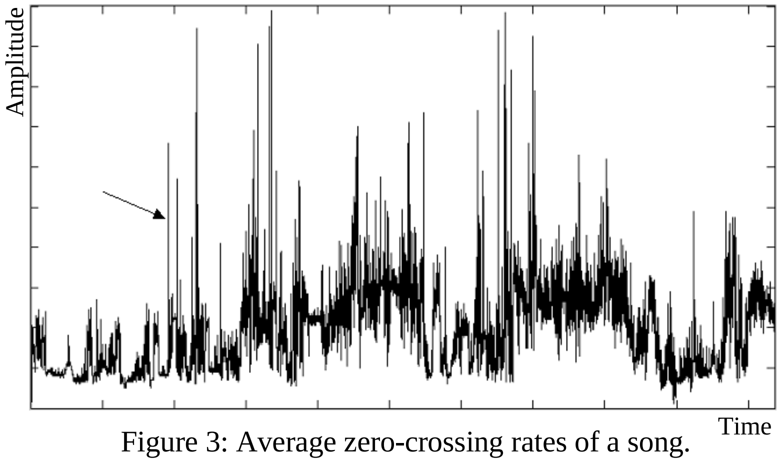


Figure 3: Average zero-crossing rates of a song.

- *The Harmonic Coefficient*: as defined in [3], the harmonic coefficient H_a is calculated by the maximum of the average autocorrelation value in the time-domain and the frequency-domain, which gives a good indication of existing harmonic components. Suppose $R^T(\tau)$ is the temporal autocorrelation for candidate pitch τ, $R^S(\tau)$ is the corresponding spectral auto-correlation, to improve robustness, combine $R^T(\tau)$ and $R^S(\tau)$ as:

  $$R(\tau) = \beta \cdot R^T(\tau) + (1-\beta) \cdot R^S(\tau) \tag{4.1}$$

  where β=0.5 turned out to perform well. Then, H_a is defined as:

  $$H_a = \max_{\tau} R(\tau) \tag{4.2}$$

  According to [3], the singing signal in general has higher values of the harmonic component, compared to the instrumental music. Therefore, the start of the singing voice may be indicated by an abrupt increase in the H_a value.

- *The Spectral Flux*: it is defined as the 2-norm of the frame-to-frame spectral amplitude difference vector:

  $$F_n = \left\| |X_n(\omega)| - |X_{n+1}(\omega)| \right\| \tag{4.3}$$

  where $|X_n(\omega)|$ is the magnitude spectrum of the nth frame of the audio signal. The start of singing voice is often indicated by the appearance of high peaks in the spectral flux value, because the voice signal tends to have higher rate of change than instrumental music. An example is shown in Fig.4, where the arrow denotes the start of the singing voice in the spectral flux values of a song.

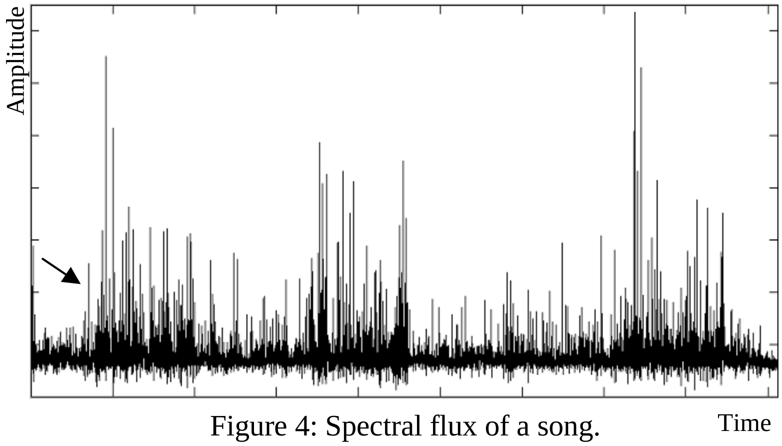


Figure 4: Spectral flux of a song.

The above four audio features are computed once every 15ms of the audio signal in the song, over a window of 20ms, and compared with a set of predetermined thresholds, until the start of the singing voice is detected. The flowchart of this procedure is plotted in Fig.5 at the end of this report. This algorithm has proved to be able to get rid of most of the prelude in songs and extract segments consisting of typical voice signals of the singer. For example, in a recording of the song “Believe” performed by Cher, there is a relatively long prelude of 30 seconds. With the developed procedure, the whole prelude was discarded and the start of the singing signal was accurately located. Once the start of the singer’s voice is determined, 25 seconds of the audio signal are excerpted from that point and used as testing data.

4.2. Matching with Singer’s Voice Model

Even within singing segments of songs, there are still moments when the instrumental sounds, rather than the singing voice, are dominant (e.g. when the singer takes a breath). For a testing segment of the song as obtained in the last step, it is divided into audio frames, and the LPC derived cepstral coefficients of each frame are computed. These audio features are matched with the GMM model of each singer by computing likelihood values according to formula (3.1). If an audio frame is dominant with a certain singer’s voice, its feature vector should match with this singer’s model with a high likelihood value and vote exclusively to this model. Otherwise, if instrumental sounds are dominant in one audio frame, its feature vector should have a low likelihood value when matched with any singer’s model, and will not vote to any of the models. Finally, the GMM model with the most frames voting to it is elected, and the song is classified to the corresponding singer.

4.3. Clustering Songs Similar in Singer’s Voice

Songs in a music collection can be categorized based on singer’s voice with a similar process as the singer identification. For this purpose, one GMM model needs to be trained for each type of singing voice, e.g. soprano, mezzo-soprano, tenor, baritone, bass, etc., using typical songs. The user may decide the granularity of the clustering. Afterwards, all songs in the music collection are matched with the models and classified to one of the clusters. An index is then assigned to each song with the proper music type.

Songs may also be retrieved from a music database which are similar to a query song in terms of singer’s voice. The user needs to manually select training segments from the query song under the guidance of an interactive user interface. Then, a model is trained for the singing voice in the query song, and songs in the database are matched with this model. Those songs with high likelihood values are considered similar to the query song.

5. Experimental Results

The above method was tested using a small collection of sample songs of eight singers: four males and four females; four English and four Chinese. For each singer, one song was selected as the training sample. And there were 45 songs in the testing sample set, with three to nine samples of each singer.

Each training sample is about one minute long which contains segments manually selected from the song to represent the voice of the singer. The LPC derived MFCC of each training sample were computed once every 20ms, and the resulted feature vector series were used to build a GMM for the singer's voice. For each testing song, the start of the singer's voice was first detected using our developed method, then the first 25 seconds of the singer's voice (normally mixed with sounds of musical instruments) was extracted and used as the testing data. Then, the MFCC feature vectors for each testing sample were computed, and they were matched with the

GMM model of each singer's voice. By comparing the likelihood values, each testing sample was classified to one singer. The results are shown in Table 1,

Table 1: Singer identification results using proposed scheme.

Singer’s name	Correct	Incorrect	Total
Andy Williams	6	0	6
Elvis Presley	1	2	3
Barbra Streisand	6	0	6
Ella Fitzgerald	9	0	9
Liu, Huan	2	3	5
Liu, Wen-Zheng	4	0	4
Deng, Li-Jun	8	0	8
Meng, Ting-Wei	2	2	4
Total	38	7	45

The sample songs are of different styles. For example, some songs have quite fast tempo while there are other songs having rather slow tempo. In some of the songs, the singer's voice is dominant in most of the song, or at least in many segments of the song; while in the rest of the songs, there is a strong instrumental background most of the time. From Table 1, we can see that the singer was correctly identified in 82% of the testing samples, which shows that the proposed method is very promising.

Nevertheless, the accuracy rate is not consistent among the singers. The most important factor is the selection of the training samples – the ideal situation is that only the singer's voice is extracted and there is no other sound existing; while in reality, the lighter the instrumental sounds, the better the results. Next, for testing samples, the more dominant the singer's voice is in the testing data, the better chance it can be correctly identified. Excellent results were achieved for the sample songs of five singers because the singer’s voice is loud and clear compared to the background music, such as in the songs of Andy Williams and Ella Fitzgerald. However, the instrumental music is rather strong in the songs of the other three singers, and a large percent of the songs were misclassified to unrelated singers. Moreover, parameters of the audio features and statistical models affect the performances to some extent as well. For instance, if the number of mixtures in the Gaussian mixture model is changed from 5 to 16, the overall accuracy rate would drop to 75%.

6. Conclusions and Future Work

We conclude that, in general, the automatic singer identification problem can be solved to a certain extent using an approach similar to speaker identification systems while special techniques are necessary to distinguish the singer's voice from the song. The background instrumental music makes the problem complicated, but with proper audio processing methods, the accuracy rate may reach an acceptable level and the proposed scheme may be used in music management systems, which will greatly improve the indexing, storing and retrieval functionalities.

To automate the selection of training samples, in the next step, we will work on detecting segments in a song in which the singing voice is dominant, based on limited prior arts available at present [4]. Furthermore, as results of the current scheme depend heavily on the amount of instrumental music involved in the training and testing samples, to improve the performance, we will investigate audio segregation theories and methods [5] to reduce the influence of the background music. More experiments will also be conducted on larger sample sets to test the identification, clustering and retrieval performances.

7. References

[1] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Inc., New Jersey, 1978.

[2] D. Reynolds and R. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” *IEEE Transactions on Speech and Audio Processing*, vol.3, no.1, pp.72-83, 1995.

[3] W. Chou and L. Gu, “Robust singing detection in speech/ music discriminator design,” *Proceedings of ICASSP’01*, vol.2, pp.865-868, Salt Lake City, May 2001.

[4] A.L. Berenzweig and D.P.W. Ellis, “Locating singing voice segments within music signals,” *Proceedings of IEEE WASPAA’01*, pp.119-122, New York, Oct. 2001.

[5] G.J. Brown and M. Cooke, “Computational auditory scene analysis,” *Computer Speech and Language*, vol.8, no.2, pp.297-336, 1994.

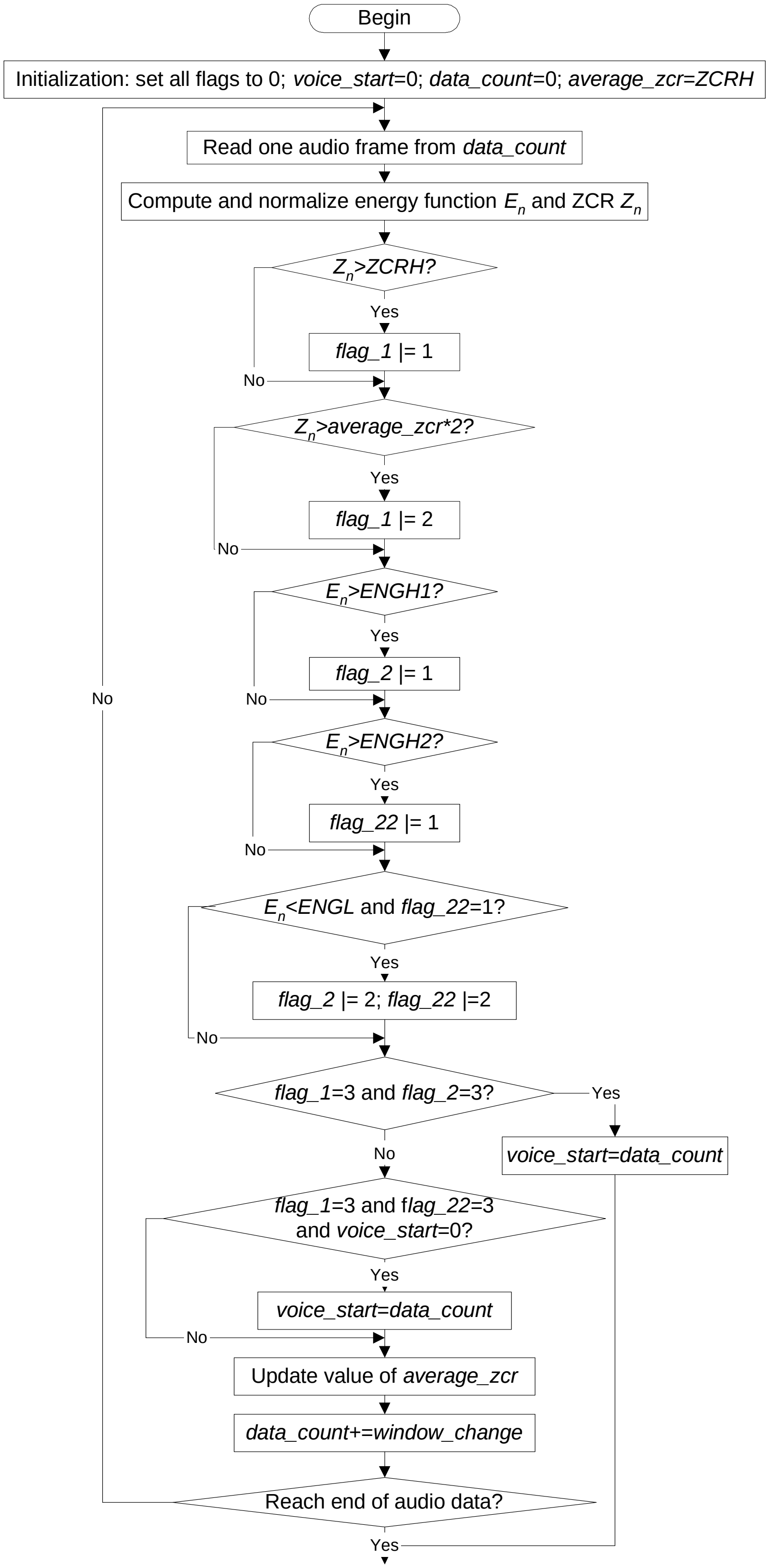

Begin
Initialization: set all flags to 0; *voice_start*=0; *data_count*=0; *average_zcr*=*ZCRH*
Read one audio frame from *data_count*
Compute and normalize energy function E_n and ZCR Z_n
Z_n>ZCRH?
Yes
flag_1 |= 1
No
Z_n>average_zcr*2?
Yes
flag_1 |= 2
No
E_n>ENGH1?
Yes
flag_2 |= 1
No
No
E_n>ENGH2?
Yes
flag_22 |= 1
No
E_n<ENGL and *flag_22*=1?
Yes
flag_2 |= 2; *flag_22* |=2
No
flag_1=3 and *flag_2*=3?
Yes
voice_start=*data_count*
No
flag_1=3 and f*lag_22*=3
and *voice_start*=0?
Yes
voice_start=*data_count*
No
Update value of *average_zcr*
data_count+=*window_change*
Reach end of audio data?
Yes

Figure 5: Flowchart for detecting the start of singing voice in a song.

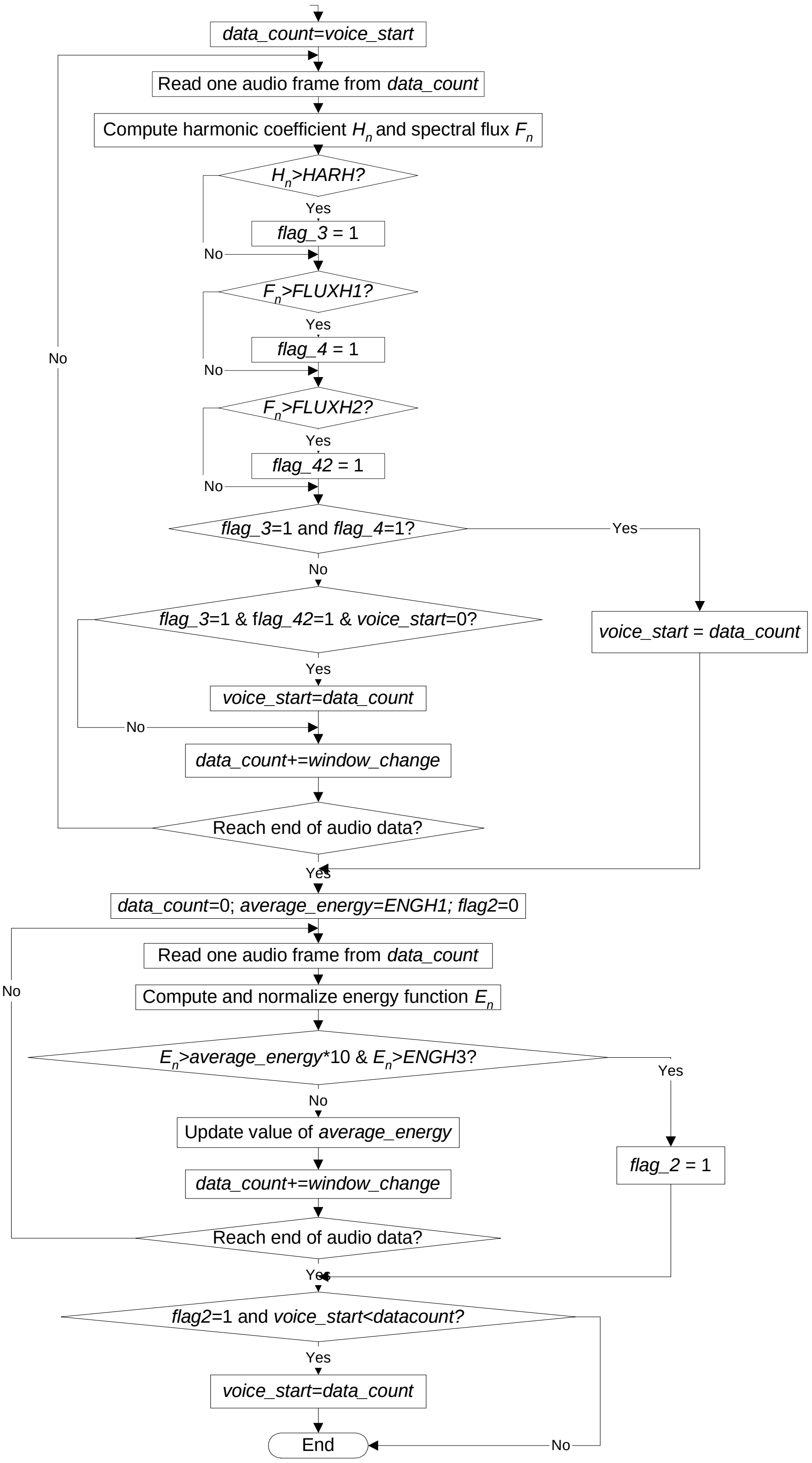

data_count=voice_start
Read one audio frame from data_count
Compute harmonic coefficient H_n and spectral flux F_n
H_n>HARH?
Yes
flag_3 = 1
No
F_n>FLUXH1?
Yes
flag_4 = 1
No
No
F_n>FLUXH2?
Yes
flag_42 = 1
No
flag_3=1 and flag_4=1?
Yes
No
flag_3=1 & flag_42=1 & voice_start=0?
voice_start = data_count
Yes
voice_start=data_count
No
data_count+=window_change
Reach end of audio data?
Yes
data_count=0; average_energy=ENGH1; flag2=0
Read one audio frame from data_count
No
Compute and normalize energy function E_n
E_n>average_energy*10 & E_n>ENGH3?
Yes
No
Update value of average_energy
flag_2 = 1
data_count+=window_change
Reach end of audio data?
Yes
flag2=1 and voice_start<datacount?
Yes
voice_start=data_count
End
No

Figure 5: Flowchart for detecting the start of singing voice in a song (continued).