Hindawi
Journal of Mathematics
Volume 2024, Article ID 9852879, 1 page
https://doi.org/10.1155/2024/9852879

Retraction

Retracted: Music Segmentation Algorithm Based on Self-Adaptive Update of Confidence Measure

Journal of Mathematics

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Li, "Music Segmentation Algorithm Based on Self-Adaptive Update of Confidence Measure," *Journal of Mathematics*, vol. 2021, Article ID 8329088, 9 pages, 2021.

Hindawi
Journal of Mathematics
Volume 2021, Article ID 8329088, 9 pages
https://doi.org/10.1155/2021/8329088

Research Article

Music Segmentation Algorithm Based on Self-Adaptive Update of Confidence Measure

Jianhua Li

School of Music and Dance, Huaihua University, Huaihua, Hunan 418000, China

Correspondence should be addressed to Jianhua Li; ljh@hhtc.edu.cn

Received 9 October 2021; Revised 23 October 2021; Accepted 27 October 2021; Published 8 November 2021

Academic Editor: Naeem Jan

To improve the accuracy of music segmentation and enhance segmentation effect, an algorithm based on the adaptive update of confidence measure is proposed. According to the theory of compressed sensing, the music fragments are denoised, and thus the denoised signals are subjected to short-term correlation analysis. Then, the pitch frequency is extracted, and the music fragments are roughly classified by wavelet transform to realize the preprocessing of the music fragments. In order to calculate the confidence measure of the music segment, the SVM method is used, whereas the adaptive update of the confidence measure is studied using reliable data selection algorithm. The dynamic threshold notes are segmented according to the update result to realize music segmentation. Experimental results show that the recall and precision values of the algorithm reach 97.5% and 93.8%, respectively, the segmentation error rate is low, and it can achieve effective segmentation of music fragments, indicating that the algorithm is effective.

1. Introduction

The audio signal in a music clip is a complex mixture of multiple sound signals (voice, music, environmental sound, etc.) intertwined. When converting from one type of audio signal to another, some auditory features will change, and the difference between the front and back is large, just like the visual features in the image sequence [1]. The purpose of music segmentation is to distinguish different audio signals according to audio characteristics in preparation for subsequent audio processing, such as classification [2]. Through music segmentation, different processing methods can be adopted for different types of audio signals to reduce the search space for further processing. Besides, the results of this segmentation reflect the high-level semantic features of audio content, especially the features of audio clips, which are of great importance for music retrieval and understanding music content [3].

At present, the commonly used music segmentation methods mainly include real-time audio segmentation method based on adaptive threshold adjustment, audio segmentation algorithm based on hierarchical entropy detection, audio segmentation algorithm based on credibility change trend, and audio segmentation algorithm based on fixed-length window hierarchical detection. Among them, the real-time audio segmentation method based on threshold adaptive adjustment mainly aims at real-time audio applications, takes the environmental factor as the measurement of external environment detection, and uses it to adaptively adjust the segmentation threshold. Finally, the table lookup method is used to judge the segmentation type through state transition to achieve a balance between efficiency and accuracy, so as to realize music segmentation. The audio segmentation algorithm based on hierarchical entropy detection uses the fixed length analysis window hierarchical structure to traverse the audio stream, and the jump points are detected according to the entropy change trend in the window. Experimental results show that the algorithm is an effective audio segmentation method. The audio segmentation algorithm based on the change trend of credibility adopts the fixed length sliding window detection structure to reduce the cumulative error, calculates the credibility of each audio frame in the window, and detects the jump point according to the change trend of credibility, so as to avoid the false detection caused by threshold selection and hard threshold decision. Experimental results show that the algorithm has good segmentation performance. The audio segmentation algorithm based on fixed-length window layered detection uses fixed-length window sliding to traverse the audio stream, and the detected jump points are calculated from top to bottom in the window. Finally, the detected candidate jump points are verified by local extreme value determination method. Experimental results show that the processing speed of this algorithm is greatly improved compared with other segmentation algorithms.

Although the above methods can realize music segmentation because the noise interference is not considered, the error rate of segmentation results is high, and the accuracy needs to be improved. Therefore, this paper proposes a music segmentation algorithm based on an adaptive update of confidence measure.

In this paper, we classify the musical fragments following some methods and theories in Section 2. Further research on the segmentation algorithm to achieve accurate music segmentation is presented in Section 3. Additionally, experimental verifications on the effectiveness of the music segmentation algorithm based on the adaptive update of the confidence measure are done in Section 4. A conclusion is presented in Section 5 that includes a summary of the methods used to solve the problem of low accuracy of music segmentation.

2. Music Preprocessing

In this section, the accuracy of segmentation is taken into consideration, where we use vocal and nonvocal attributes of music segments. To avoid the noise signals that occur in this situation, the compressed sensing theory is used. The classification of music by wavelet transform to realize if the music fragments are ready to be processed is also presented.

2.1. Music Segment Presegmentation. Music signal has short-term stationary characteristics; that is, the signal characteristics are basically stable in a limited short time period. According to different music signal characteristics, the duration of stationary segments ranges from hundreds of milliseconds to several seconds. Segment presegmentation is to divide the continuous nonstationary signal flow into a series of short-term stationary segments. At present, the commonly used presegmentation methods can be divided into fixed-length segmentation and indefinite length segmentation. The former directly divides the processed signal into several equal length segments according to the characteristics of the processed signal and assumes that each segment contains only one kind of sound source. In the latter, the whole signal stream is predivided into several short segments by the signal spectrum breakpoint detection algorithm. These two presegmentation algorithms are difficult to avoid presegmentation errors; that is, a segment may contain different sound sources, resulting in segmentation errors.

In an effort to become more precise, this paper used to break music segments into small forms of vocal and nonvocal model training because of their quality that directly affects the accuracy of music segment presegmentation. Suppose that there is a set $A = \{a_1, a_2, \dots, a_n\}$ such that a_i either belongs to a vocal or to a nonvocal music segment. Let $\{b_{11,} b_{12}, \dots, b_{1m}\}$ be the m-frame feature vector of the a_i segment. φ_c and φ_{nc} are the vocal and nonvocal presegmentation models, respectively. The log likelihood rates of the two are

$$\begin{aligned}\varphi_c(a_i) &= \text{net}_i(k) \times m_1(k) \quad i = 1, 2, \dots, N,\\ \varphi_{nc}(a_i) &= \text{net}_i(k) \times m_2(k) \quad i = 1, 2, \dots, N.\end{aligned} \tag{1}$$

In the previous formula, $\text{net}_i(k)$ represents the music piece to be divided; $m_1(k)$ represents the vocal music model that matches the music piece to be divided; $m_2(k)$ represents the nonvocal music model that matches the music piece to be divided. If $\varphi_c(a_i) > \varphi_{nc}(a_i)$, then a_i is vocal music; otherwise, it is nonvocal music.

Due to the many types and genres of music and songs, the instruments used, the way of playing, the singer's voice, and singing are all very different. In order to fully characterize different music characteristics as much as possible, a large amount of music is often needed as training data to establish φ_c and φ_{nc}. On the one hand, this increases the complexity of the model. On the other hand, it increases the mismatch between the model and a particular piece of music to be segmented. Therefore, reducing the complexity of the model while making it closer to each piece of music to be processed is the key to improving the segmentation accuracy.

2.2. Music Signal Denoising. According to Section 2.1, the vocal and nonvocal attributes of music segments can be preliminarily obtained. Although it provides a certain basis for music segmentation, due to the influence of external interference conditions, there will be a lot of noise in music segments [4], which will affect the effect of music segmentation. Therefore the compressed sensing theory will be used to denoise the noise signals in music segments [5].

The core idea of compressed sensing theory [6] is to use nonadaptive linear projection of the collected signal and then reconstruct the original signal from the measured value according to the corresponding reconstruction algorithm. Given a measurement matrix as $\Psi \in E^{m\times n}$ $(m \ll n)$, and define the linear measurement value as F, $F \in E^m$, of the signal $f(n) \in E^n$ under the measurement matrix, namely:

$$F = \Psi f(n). \tag{2}$$

Now consider reconstructing $f(n)$ from F. Obviously, since the dimension of F is much lower than that of $f(n)$, the above equation has infinite solutions. The theory proves that the signal $f(n)$ can be accurately calculated from the measured value F by solving the optimal norm. Refactor

$$\widehat{F} = \arg\min\|F\| \tag{3}$$

or

$$\widehat{F} = \arg\min\|F - \Psi f(n)\|_2^2. \tag{4}$$

On the basis of signal reconstruction, considering that the wavelet coefficients of noise signals at various scales do not have sparseness [7], compressed sensing theory can be used to restore the sparseness of wavelet coefficients, so as to achieve the purpose of signal noise reduction. The denoising process is shown in Figure 1.

In the denoising process, the random measurement matrix Ψ needs to satisfy the principle of uniform uncertainty, that is, for any sparse vector h, if

$$0.6\frac{M}{N}\|h\|_2^2 \le \|\Psi h\|_2^2 \le 1.4\frac{M}{N}\|h\|_2^2. \tag{5}$$

It is said that $\Psi_{M\times N}$ satisfies the principle of unanimous uncertainty and the music signal can be denoised.

2.3. Pitch Frequency Extraction. Perform short-term correlation analysis on the denoised music signal, and define the autocorrelation function of a certain frame of signal as

$$x_i(k) = \sum_{i=1}^{N} u(k)u_i(k+g). \tag{6}$$

In the formula, $u(k)$ represents a certain music signal, $u_i(k)$ represents a section of windowed and framed signal, and g represents a lag.

The autocorrelation function will have a peak at an integer multiple of the pitch period, so the pitch period value can be extracted by detecting the position of the peak. To ensure that the pitch is extracted correctly, the window length is set to be greater than 2 pitch periods when framing, and the median smoothing method is used to remove the "outliers" caused by the pitch extraction process:

$$R(n) = w(n) - r(m) \times p_i. \tag{7}$$

In the formula, $w(n)$ represents the input signal, $r(m)$ represents the output of the median filter, and p_i represents the smoothing window, which satisfies

$$\sum_{i=1}^{l} p_i = 1. \tag{8}$$

In the formula, l represents the length of the smooth window.

The pitch of each frame of the music signal is extracted to form a pitch frequency string. Since the change of the pitch string value can correspond to the pitch change, it is convenient to search and classify music according to the pitch change.

2.4. Rough Classification of Music Fragments. In the rough classification of music fragments, a comprehensive processing method is adopted in this paper. This method combines gene recognition and chord recognition in an algorithm to process simultaneously, which improves computational efficiency [8]. The basic idea is to first analyze the music data structure through wavelet transform, then screen the part with the largest amplitude as the comparison item, and time the pronunciation time of adjacent candidate items. Finally, compare the adjacent data one by one to find out the large amplitude change point and record the duration of large and small amplitude domain; thus, the desired rough classification effect can be obtained.

Figure 2 is a schematic diagram of the frequency-amplitude of the note comparison item of a mixed note measure. Among them, F_1 is the single note comparison item with frequency t_1, Y_1 is the amplitude; F_2, F_3, and F_4 are the triad note comparison items with the frequency of the consonant components being t_2, t_3, and t_4, respectively, and Y_2, Y_3, and Y_4 are the corresponding amplitudes; F_5 is a single note comparison item with frequency t_5, and Y_5 is its amplitude. The elements with the smaller amplitude in the middle are not marked. These elements have little significance for the rough classification of music and can be ignored by the given selection threshold. The frequency-amplitude diagram of the note comparison item is shown in Figure 2.

In order to achieve a rough classification of music, a number sequence $W_j = \{w_{j1}, w_{j2}, \ldots, w_{jn}\}$ can be defined, where w_{ji} represents the note included in the selected j-th note comparison item. If it is a single note, then $i = 1$. If it is a chorus, then $i = n$. In addition, a number sequence $D_{W_j} = \{d_{wj1}, d_{wj2}, \ldots, d_{wjn}\}$ can be defined, where d_{wji} represents the note strength contained in the filtered j-th note comparison item, $j = 1, 2, \ldots, i = 1, 2, \ldots$. Let the comparison coefficient be D_j, and its calculation formula is

$$D_j = \frac{D_{z+1}}{D_z} \quad z = 0, 1, 2, \ldots. \tag{9}$$

In the formula, D_z represents the average energy of the z-th note comparison item. The note comparison item can be a single sound or a consonant. The expression is

$$D_z = \overline{D_{W_j}} = \sum_{j=1}^{n} \frac{d_{wji}}{n}. \tag{10}$$

When the note comparison item is monophonic, $j = n = 1$, its balance energy $D_j = D_{W_j} = D_{wj1}$.

The value of D_j can be used to judge the changes in the comparison items of adjacent notes. If the value of D_j is in the closed interval [0.6, 1.4] (the value is an empirical value), the change can be approximated as a small change in the same set of coarse emotional domains. If the value of D_j exceeds this range, its change can be approximated as a jump in the coarse emotional domain. However, a common situation arises in this comparison, namely,

$$\begin{gathered} d_1, d_2, \ldots, d_{j-1} \in [0.6, 1.4] \\ d_j, d_{j+1}, \ldots, d_{j+m} \notin [0.6, 1.4] \\ d_{j+m+1}, \ldots \in [0.6, 1.4]. \end{gathered} \tag{11}$$

For example, $m = 1$ or $m = 2$; this occasional single or several jumps, based on experience, are not enough to show

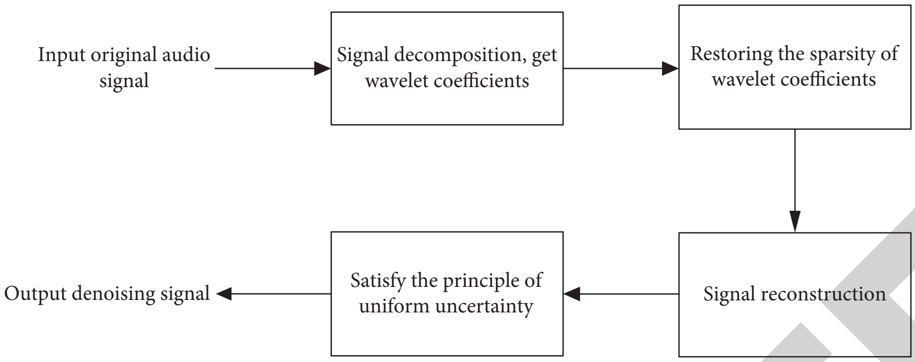
Input original audio signal
Signal decomposition, get wavelet coefficients
Restoring the sparsity of wavelet coefficients
Signal reconstruction
Satisfy the principle of uniform uncertainty
Output denoising signal

Figure 1: Music denoising process.

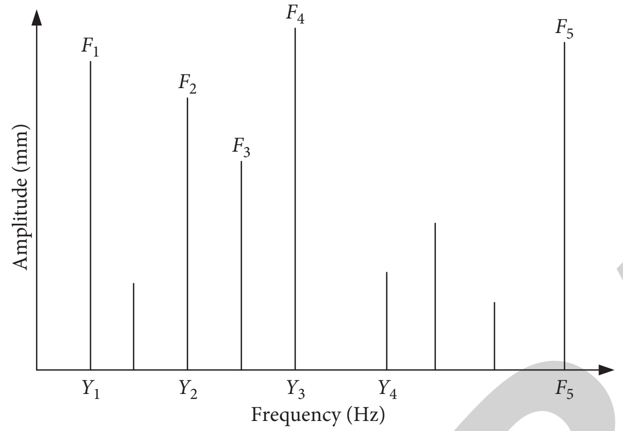

Figure 2: The frequency-amplitude diagram of the note comparison item.

that the emotion of music has jumped in different coarse emotional domains, so certain restrictions on m need to be made. m can be set within a certain range according to the actual situation of the music, so as to ensure that the music with emotional changes can stably stay in its emotional domain for a period of time, so that the emotional information of this music segment can be fully displayed. Otherwise, it needs to be regarded as an invalid emotional expression segment, so the whole music can be roughly classified according to the local intensity and rhythm of the music.

3. Realization of Music Segmentation

The presegmentation of music segments, music signal denoising, pitch frequency extraction, and rough classification of music segments are realized through the preprocessing of music segments. The processing results provide a solid foundation for music segmentation. On this basis, in order to achieve accurate music segmentation, further research is conducted on the segmentation algorithm.

3.1. Music Segment Processing Based on Adaptive Update of Confidence Measure

3.1.1. Calculation of Confidence Measure of Music Fragment. In this paper, the SVM method [9] is used to calculate the confidence measure of music fragments. First, the confidence measure feature vector is extracted for each syllable according to the information in the candidate. Assuming that μ is a keyword composed of N_w syllables, its corresponding candidate μ_w can be decomposed for N'_w syllable candidates, and syllable v_i^w corresponds to candidate $\mu_{v_i^w}$. The SVM classifier is used to obtain the classification score $S(\mu_{v_i^w})$ of the confidence measure feature vector corresponding to the candidate $\mu_{v_i^w}$, and then the confidence measure $S_{\mathrm{SVM}}(\mu_{v_i^w})$ of the syllable candidate is calculated according to the classification score.

The specific calculation steps are as follows: first, use the Sigmoid function to normalize the score of the SVM classifier to (1, 1); then, take the logarithm of the normalized result. The above process is similar to the process of calculating factor-level confidence measures. The specific calculation formula is

$$S_{\mathrm{SVM}}\left(\mu_{v_i^w}\right) = \ln \frac{1}{1+\exp\left(\vartheta \times S\left(\mu_{v_i^w}\right)\right)}. \qquad (12)$$

In the formula, ϑ represents a constant that controls the smoothness of the Sigmoid function.

Next, according to the classification score of each syllable candidate, the confidence measure of the entire keyword candidate can be calculated. The simplest way is to use the classification score to take the average [10]:

$$\overline{S} = \frac{1}{\mu_{v_i^w}} \sum_{i=1}^{n} S\left(\mu_{v_i^w}\right). \qquad (13)$$

As with the phoneme-based weighted confidence measure, if the syllable-level confidence measure distribution is considered different, different weights can be used to weigh the syllable-level confidence measure. As a result, a syllable-based weighted confidence measure is obtained, namely,

$$O_{\mathrm{SVM}}\left(\mu_{v_i^w}\right) = \frac{1}{\mu_{v_i^w}}\left[\alpha_i + \beta_j\right]. \qquad (14)$$

In the formula, α_i and β_j, respectively, represent the linear weighting coefficient and the offset. These two parameters can be obtained according to the weighting coefficient training method.

3.1.2. Confidence Measure Adaptive Update Data Selection. Due to the different training data, the pretrained vocal and nonvocal models cannot accurately characterize the acoustic characteristics of the music signal to be segmented. The mismatch between the model and the processed data will lead to serious errors in the segmentation results. If vocal and nonvocal signals are extracted from the music signals to be segmented and the corresponding model is updated adaptively, the matching degree between the model and the processed data can be improved. Since the vocal and nonvocal parts of music signals are known, a reliable data selection algorithm is proposed for adaptive updating of confidence measure.

Based on the presegmentation of music clips, use φ_c and φ_{nc} to divide music clips into two categories, namely,

$$\begin{aligned}\widehat{R}_{ri} &= \varphi_c\left(W_1^0, W_2^0\right),\\ \widehat{R}_{rj} &= \varphi_{nc} \times \overline{R}_{ij}\left(W_j^0\right).\end{aligned} \tag{15}$$

In the formula, W_1^0 represents the pure music and voice segments, W_2^0 represents the music mixed segments, and W_j^0 represents the music climax segments.

Due to the mismatch between the model and the processed data, there are misidentified data in $\widehat{R}_{ri}$ and $\widehat{R}_{rj}$, that is, fragments containing opposite classes. Confidence measures δ_1 and δ_2 are used to judge the reliability of the segments in $\widehat{R}_{ri}$ and $\widehat{R}_{rj}$, respectively, which are defined as

$$\begin{aligned}\delta_1 &= \frac{1}{\mu_1}\frac{\left|\omega_{r1}(k)\right|^2}{\sigma_1(k)^2},\\ \delta_2 &= \frac{1}{\mu_2}\frac{\left|\omega_{r2}(k)\right|^2}{\sigma_2(k)^2}.\end{aligned} \tag{16}$$

The larger the δ_1 and δ_2, the greater the possibility that the segment is correctly identified as vocal or nonvocal. Research shows that δ_1 and δ_2 are approximately normally distributed. The reliable data selection criteria adopted in this study are as below.

Let χ_1 and χ_2 be the mean and standard deviation of δ_i, respectively. For each segment W_{ij}, if its confidence measure $\delta_i > \chi_1 - l\chi_2$, then W_{ij} is reliable data, which can be used for the model update. Through analysis, it is found that the optimal segmentation result can be obtained when the value of l is around 1. Therefore, this paper takes $l = 1$ and then updates the confidence measure δ_i to obtain the data update result.

3.2. Dynamic Threshold Note Segmentation. The traditional amplitude difference note segmentation algorithm will affect the average value of the fundamental frequency due to the inaccurate calculation of the note occupancy frame or affect the number of notes due to the wrong segmentation and thus affect the segmentation accuracy. Therefore, based on the adaptive update of the confidence measure, this paper uses the amplitude difference function to dynamically set the threshold to obtain the position of the segmentation line and set the constraint conditions to determine the segmentation line to improve the adaptability and accuracy of the algorithm.

3.2.1. Determine the Split Point. Scan the category label sequence obtained by the rough classification to find all adjacent point pairs with different categories. Each such point pair corresponds to an audio clip with a length of 12 s [11]. There is a suitable segmentation point, which is called the segmentation point boundary area.

In the boundary area of a segmentation point, the problem of accurately locating the segmentation point can be transformed into a series of two types of audio classification problems at a small scale. The specific category is determined by the category transition point pair (p, q). The boundary region of the segmentation point is divided into several continuous small audio clips, and a 17-dimensional feature vector is extracted from each small audio clip. Each small audio clip is divided into class p or class q by the corresponding two kinds of classifiers. The length of each small audio clip is 1 second, and there is no overlap between adjacent clips. After classifying these small audio clips one by one, a category label sequence about the boundary area of the segmentation point will be obtained, in which there are only category labels of class p and class q.

After adjusting the correction rules, in this category label sequence, all reasonable category jump point pairs will be used as the final segmentation point decision point pair. The so-called "reasonable" category jump point pair means that if there is a category jump point pair (p, q) in the rough segmentation sequence, those point pairs consistent with the category jump direction determined in the rough segmentation are reasonable in the process of determining the last segmentation point accordingly.

In order to ensure the accuracy of segmentation points, multiple final segmentation points are allowed in the boundary area of a segmentation point. If a reasonable category jump point pair cannot be found in the boundary region of a segmentation point, the algorithm will give up locating the segmentation point in the boundary region of the segmentation point, which may remove some false segmentation point boundary regions determined by the coarse segmentation algorithm.

3.2.2. Amplitude Difference Function. The amplitude of the music signal will change drastically over time; especially, the amplitude of the note segmentation has a significant gap. The amplitude function in the traditional segmentation algorithm is defined as follows:

$$F(e) = \sum_{i=1}^{n} a_i(\alpha) W^i. \tag{17}$$

In the formula, $F(e)$ represents the waveform amplitude function, $a_i(\alpha)$ represents the amplitude of the sampling point, and W^i represents a certain frame of the input signal. Then, the amplitude difference function of $F(e)$ is

$$O_{F(e)} = F(e+1) - F(e). \tag{18}$$

Applying $O_{F(e)}$ is more obvious than applying $F(e)$ alone to the dividing line of a single note, which is convenient for subsequent processing.

3.2.3. Determination of Dividing Line. In order to find the dividing line, a threshold κ must be determined (usually, κ is a percentage of the mean $O_{F(e)}$); τ_i is the note start frame, τ_j is the note end frame [12], and the expressions of the two are

$$\begin{aligned} \tau_i &= \operatorname*{arg}_{O_{F(e)} \geq \kappa} O_{F(e)}, \\ \tau_j &= \operatorname*{arg}_{O_{F(e)} < \kappa} O_{F(e)}. \end{aligned} \tag{19}$$

After analysis, it can be seen that, after fixing κ, the adaptability to different music segments is poor. Even if κ changes with $O_{F(e)}$, since only the overall signal properties are considered and the local peak characteristics are ignored, errors will inevitably be caused. Therefore, each segment is calculated after a fixed percentage of the mean of the notes, and a set of dynamically changing κ is obtained:

$$\kappa = \Delta L \times \frac{\lambda_l}{L(x)}. \tag{20}$$

In the formula, ΔL represents the step length of the change and λ_l represents a certain segment divided by the variable step length.

Since κ changes in a local range, a set of dynamic segmentation values can be obtained.

In a cluster of dividing lines, there is only the only optimal division. According to the two characteristics of the note, the finiteness of the note length and the fixedness of the music rhythm, the restriction conditions are set and the optimal solution is found [13]. The judgment conditions are as follows:

(1) The number of frames occupied by notes is appropriate. The minimum and maximum frames are set through experiments to remove that obviously inappropriate segmentation.

(2) Note segmentation is uniform. Because the beat of a piece of music is certain, the notes segmented by the segmentation line are uniform. When a single note is greater than or less than 1.5 times the number of frames occupied by the adjacent note, the division is determined to be invalid.

3.3. The Realization Process of Music Segmentation. In the process of music segmentation, the length of the audio file has certain restrictions on the segmentation result. When the audio is too long, the segmentation process consumes more time. Considering the possibility of data loss due to segmentation failure, the segmented speech segment should not be too long; when the segmented speech segment is too short, it will increase the number of times the segmentation engine is called and reduce the segmentation efficiency. This article chooses to divide the length of the audio file to about 30 s. If the audio is directly divided into a length of 30 s, it will cause part of the music and voice to be divided into the same segment, which will interfere with the segmentation of the audio segment, which will inevitably cause data loss or music segmentation errors. In this paper, when audio segmentation is performed, the coarse segmentation is performed first; that is, the audio length after the initial segmentation is set to more than 5 s, and then the audio files are merged after the segmentation to obtain an audio file that meets the needs and has an appropriate length. The algorithm is summarized as follows:

Table 1: Scale and frequency comparison.

Scale	1	2	3	4
Notation	1	2	3	4
Frequency	260	279	324	337
Scale	$g1$	$a1$	$b1$	$d1$

(1) Read in the audio file and normalize the audio [14].

(2) Filter the normalized signal [5].

(3) Starting from the starting position of the audio stream, find the audio segment with signal strength greater than 0.2 and length greater than 0.3 s. If it exists, record the start and end positions of the audio segment, enter step 4, if it does not exist, the audio segment is music or noise, and end the process.

(4) Framing, adding windows, seeking short-term energy.

(5) Calculate the average short-term energy of the mute section from the start and end positions in step 3.

(6) The audio stream is finely divided, and the duration of the silent segment between the two audio segments is set to 0.2 s in this article.

(7) Calculate the effective segment ratio and silent ratio of each audio segment after segmentation, find the classification factor value, determine the type of each audio segment based on this value, and finally realize the music segmentation.

4. Simulation Experiment

In order to verify the effectiveness of the music segmentation algorithm based on the adaptive update of the confidence measure, simulation experiments are carried out.

4.1. Experimental Parameter Settings. The proposed algorithm is tested. In the experiment, the input music signal is sampled in 11.025 kHz/8 bits/monc format through a microphone and filtered by a bandpass filter. The upper and lower cut-off frequencies are fH = 3400 Hz and fL = 60 Hz–100 Hz, respectively. Use a first-order digital filter $H\,(Z) = 1 - \mu z^{-1}$ to perform high-frequency enhancement processing on the humming signal, where the value of μ is 0.98. Using Hamming window to window and frame the humming signal segment, the window length is 128, and the overlap length between frames is set to 64. The segmentation algorithm proposed in this paper is tested in an audio stream containing 4 types of audio (piano music 1, symphony 2, Beijing opera 3, pop song 4) with a total length of 1 hour. There are 56 real segmentation points in this audio stream. Table 1 is a comparison of audio stream scale and frequency.

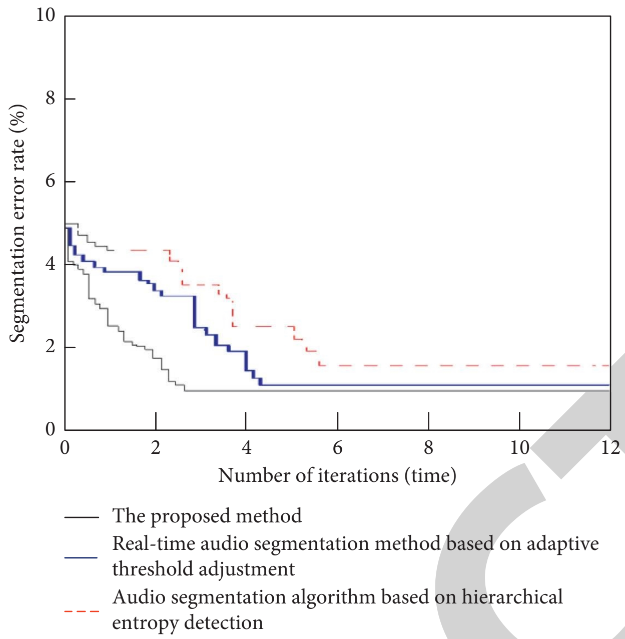

Figure 3: Comparison of music segmentation error rate.

Table 2: Comparison of recall and precision values.

Index	Method of this article	Real-time audio segmentation method based on adaptive threshold adjustment	Audio segmentation algorithm based on hierarchical entropy detection
Recall value	97.5	87.3	80.5
Precision value	93.8	85.6	79.9

Based on the information shown in Table 1, the real-time audio segmentation method based on adaptive threshold adjustment and the audio segmentation algorithm based on hierarchical entropy detection are used as comparison methods to compare with the method in this paper. The results are analyzed as follows.

4.2. Experimental Results and Analysis

4.2.1. Music Segmentation Error Rate (%). Figure 3 shows the corresponding segmentation error rates after different methods of segmentation, where the error rate is defined as the percentage of the length of the music signal that is incorrectly segmented to the total signal length.

It can be seen from Figure 3 that, as the number of iterations increases, the music segmentation error rate of different methods generally shows a trend of first decline and then stable change. Among them, the music segmentation algorithm proposed in this paper based on the adaptive update of the confidence measure reaches 3 iterations. After the second time, the error rate of music segmentation was significantly reduced. It not only has advantages in the iterative cycle but also has more obvious advantages in the error rate method of music segmentation, indicating that the segmentation results of the method in this paper are more reliable. However, the real-time audio segmentation method based on adaptive threshold adjustment and the audio segmentation algorithm based on hierarchical entropy detection has consistently higher music segmentation error rates than this method, and the segmentation effect is not good.

4.2.2. Comparison of Recall and Precision Values. In order to further verify the accuracy of the segmentation of the method in this paper, the recall and the precision values are used as comparison indicators, and the three methods are further compared and analyzed. The calculation formulas for the two parameters are

$$\begin{aligned} \text{recall} &= \frac{N_d}{N_d + N_m}, \\ \text{precision} &= \frac{N_d}{N_d + N_k}. \end{aligned} \tag{21}$$

In the formula, N_d represents a correctly segmented audio scene, N_m represents a missing segmented audio scene, N_k represents an incorrectly segmented audio scene.

Obtain the recall and the precision values according to the above formula, and the results are shown in Table 2.

According to the data in Table 2, the recall and precision values of this method reached 97.5% and 93.8%, respectively, while the recall and precision values of the real-time audio segmentation method based on adaptive threshold adjustment are 87.3% and 85.6%, respectively. Lower than the method in this paper, the audio segmentation algorithm

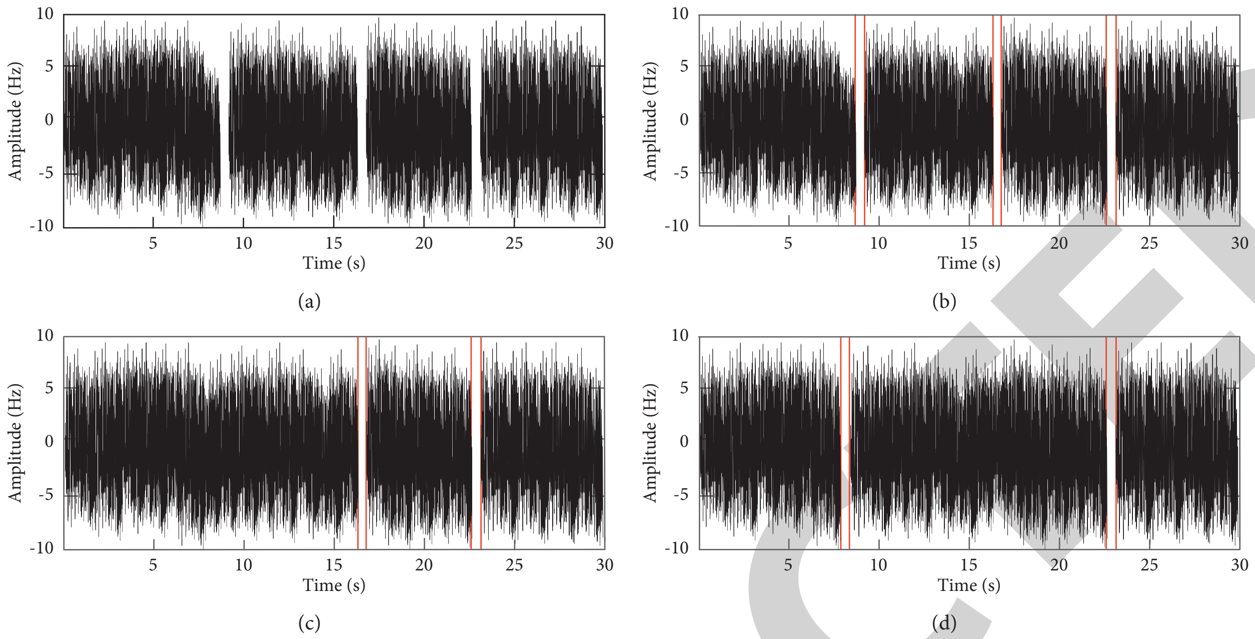

Figure 4: Comparison of segmentation results. (a) Original waveform. (b) The method of this paper. (c) Real-time audio segmentation method based on adaptive threshold adjustment. (d) Audio segmentation algorithm based on hierarchical entropy detection.

based on hierarchical entropy detection has lower recall and precision values. It can be seen that the segmentation results of this method have high accuracy, which shows the effectiveness of the segmentation method.

4.2.3. Audio Clip Segmentation Effect. Choose a piece of audio arbitrarily; the audio has pauses at 9 s, 16 s, and 23 s. Three methods are used to segment it, and the segmentation results obtained are shown in Figure 4.

By analyzing Figure 4, it can be seen that the audio file can be accurately segmented in the 9 s, 16 s, and 23 s by the proposed method, and three pause points can be obtained. The real-time audio segmentation method based on adaptive threshold adjustment can only segment the 16 s and 23 s pauses. The audio segmentation algorithm based on hierarchical entropy detection can only segment the 23 s pause, and there is a segmentation error; that is, the segmentation is performed in the 8 s, but in fact, there is no pause at that time. Therefore, the segmentation effect of this method is better, which shows that its application value is higher.

5. Conclusion

In order to solve the problem of low accuracy of music segmentation and poor segmentation effect in traditional methods, a music segmentation algorithm based on a self-adaptive update of confidence measure is proposed. The following is a summary of the innovative points of the methods in this article:

(1) Denoise the music fragment based on the theory of compressed sensing and performing short-term correlation analysis on the denoised music signal to obtain the pitch frequency

(2) Use the wavelets transform method to roughly classify the music fragments, obtain the classification results, and realize the preprocessing of the music fragments

(3) Use the SVM method to calculate the confidence measure of the music segment, and adaptively update the confidence measure

(4) According to the update results, the dynamic threshold notes are segmented to achieve music segmentation

The analysis of the experimental results shows that compared with the traditional method, the segmentation effect of the algorithm is better, which is specifically manifested in the recall and precision values, the segmentation error rate, and the segmentation effect, which fully verifies the practical application value of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding this paper.

References

[1] S. Sun, "A college music teaching system designed based on android platform," *Scientific Programming*, vol. 2021, Article ID 7460924, 16 pages, 2021.

[2] H. Purwins, B. Sturm, B. Li, J. Nam, and A. Alwan, "Introduction to the issue on data science: machine learning for

audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 203–205, 2019.

[3] C. Maximo and R. Sandra, "SART3D: a MATLAB toolbox for spatial audio and signal processing education," *Computer Applications in Engineering Education*, vol. 27, no. 4, pp. 971–985, 2019.

[4] Y. Xu, "Systematic study on expression of vocal music and science of human body noise based on wireless sensor node," *Mobile Information Systems*, vol. 2021, Article ID 9993019, 9 pages, 2021.

[5] X. Yu, Q. Peng, L. Xu, F. Jiang, J. Du, and D. Gong, "A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm," *Information Processing & Management*, vol. 58, no. 6, Article ID 102691, 2021.

[6] X. Xia and J. Yan, "Construction of music teaching evaluation model based on weighted naïve bayes," *Scientific Programming*, vol. 2021, Article ID 7196197, 9 pages, 2021.

[7] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4696–4706, 2019.

[8] S. Sun, "Evaluation of potential correlation of piano teaching using edge-enabled data and machine learning," *Mobile Information Systems*, vol. 2021, Article ID 6616284, 11 pages, 2021.

[9] Z. Yu, L. Li, W. Zhang, H. Lv, Y. Liu, and U. Khalique, "An adaptive EEG feature extraction method based on stacked denoising autoencoder for mental fatigue connectivity," *Neural Plasticity*, vol. 2021, Article ID 3965385, 12 pages, 2021.

[10] F. Cheng, H. Zhang, D. Yuan, and M. Sun, "Leveraging semantic segmentation with learning-based confidence measure," *Neurocomputing*, vol. 329, no. 2, pp. 21–31, 2019.

[11] M. Q. . Sun, "Simulation of digital audio music recognition based on time-frequency domain information extraction," *Computer Simulation*, vol. 38, no. 7, pp. 415–418, 2021.

[12] T. Popescu, R. Widdess, and M. Rohrmeier, "Western listeners detect boundary hierarchy in Indian music: a segmentation study," *Scientific Reports*, vol. 11, no. 1, p. 3112, 2021.

[13] Y. Xu, J. Yang, and K. Mao, "Semantic-filtered Soft-Split-Aware video captioning with audio-augmented feature," *Neurocomputing*, vol. 357, no. 9, pp. 24–35, 2019.

[14] K. Zhang, M. J. Sjerps, and G. Peng, "Integral perception, but separate processing: the perceptual normalization of lexical tones and vowels," *Neuropsychologia*, vol. 156, no. 1, Article ID 107839, 2021.