J. Shanghai Jiaotong Univ. (Sci.), 2014, 19(4): 392-397
DOI: 10.1007/s12204-014-1515-5

Audio Authenticity: Duplicated Audio Segment Detection in Waveform Audio File

XIAO Ji-nian[1] (肖佶年), *JIA Yun-zhe*[1] (贾蕴哲), *FU Er-dong*[1] (付尔东)
HUANG Zheng[1*] (黄　征), *LI Yan*[2] (李　岩), *SHI Shao-pei*[2] (施少培)
(1. College of Information Security, Shanghai Jiaotong University, Shanghai 200240, China;
2. Institute of Forensic Science, Ministry of Justice, Shanghai 200063, China)

Abstract: Waveform audio (WAV) file is a widely used file format of uncompressed audio. With the rapid development of digital media technology, one can easily insert duplicated segments with powerful audio editing software, e.g. inserting a segment of audio with negative meaning into the existing audio file. The duplicated segments can change the meaning of the audio file totally. So for a WAV file to be used as evidence in legal proceedings and historical documents, it is very importance to identify if there are any duplicated segments in it. This paper proposes a method to detect duplicated segments in a WAV file. Our method is based on the similarity calculation between two different segments. Duplicated segments are prone to having similar audio waveform, i.e., a high similarity. We use fast convolution algorithm to calculate the similarity, which makes our method quit efficient. We calculate the similarity between any two different segments in a digital audio file and use the similarity to judge which segments are duplicated. Experimental results show the feasibility and efficiency of our method on detecting duplicated audio segments.

Key words: duplicated audio segment, similarity, convolution
CLC number: TP 309.2　　**Document code:** A

0 Introduction

Today people have more concern about the authentication of digital audio as the audio editing software is becoming more powerful. There are many ways to tamper digital audio. Audio forensics is becoming more and more important as digital audio can be used as evidence in court and other special occasions.

The duplicate audio inserting is one of the methods to falsify digital audio. Suppose that we have someone read several sentences and record the voice into waveform audio (WAV) file format. One can identify a segment of audio wave for the word "not" and insert the segment into any sentence he wants to change the meaning. This kind of modification could bring trouble in digital audio forensic. It, no doubt, can lead to unanticipated consequences once some part in audio file being duplicated maliciously.

The authenticity of an audio file has got some researchers' attention recently as the authenticity of digital audio becomes more and more important. And inserting duplicated segment into WAV file is a common audio tampering method. There are indeed some technologies to detect the authenticity of digital audio, but little effort has been put to the research on the detection of duplicated segment.

The researches of audio forensics start from 1990s. There have been many achievements. Farid[1] used bispectral analysis to detect digital forgery in speech signal on the basis of the assumption that a "natural" signal has weak higher-order statistical correlations in the frequency domain, and forgery in speech would introduce "un-natural" correlations. Cano et al.[2] brought on an audio fingerprinting system for duplicate detection. Grigoras[3] pointed out that digital equipment captured the intended speech and the 50/60 Hz electric network frequency (ENF) when recording. The ENF criterion could be used to check the integrity of digital audio recordings and to verify the exact time when a digital recording was created. Duplicate song detection using audio fingerprinting for consumer electronics devices also showed up in 2006[4]. Yao et al.[5] utilized expectation-maximization (EM) algorithm in parameter estimation of periodic linear correlation to detect whether the signal is interpolated. This method is confined to linear interpolation detection. Kraetzer et al.[6] proposed a method to determine the authenticity of the

Received date: 2013-10-30
Foundation item: the "12th Five-Year Plan" National Science and Technology Support Program (No. 2012BAK16B05)
***E-mail:** huang-zheng@sjtu.edu.cn

speaker's environment. It was said that the extraction of background features of an audio stream could provide an informative basis for determining its origin location and the used microphone. Yang et al.[7] introduced a format conversion dependent method for locating forgeries (insertions and deletions) in MP3 files by time domain based analysis of encoder frame offsets. Maher[8] provided the forensic examination which has presented an overview of current practices in the field of audio forensics. Meanwhile, Maher[9] gave an overview about the relevant technology about audio and reviewed several areas for the research and development in 2010. Rodríguez et al.[10] presented audio authenticity about detecting ENF discontinuity with high precision phase analysis. For the problem of detecting duplicated segment, little effort can be found on this matter.

The duplicated segment in WAV file has posed a threat to the authenticity of digital audio. This paper proposes a method to detect duplicated segments in an audio file. Based on the essential characteristics of WAV file, this paper proposes a concept of similarity which is related to covariance function. We can locate the duplicated segment through calculating the similarity between two different parts in an audio file using convolution. We set a threshold for similarity. The segments will be considered duplicate if the value of similarity is above the threshold. This method also uses the technology about fast Fourier transform algorithm; with it, the execution time can be dramatically reduced without changing the results.

1 Similarity Between Audio Segments

The purpose of this section is to define a concept of similarity which is used to describe the similar degree between two segments of an audio file, and how to calculate the similarity between the two parts.

1.1 Definition of Similarity

From the basic characteristics about audio signal, it is obvious that the similar parts of audio are bound to have similar audio signals. We define the similarity which represents the similar degree between two different segments in one audio file. Based on this similarity property, this paper proposes a method to detect duplicate audio through computing the similarity between every two segments in the audio file, and gives a threshold to determine whether the audio file has duplicated segment inserted. The definition of similarity is as follows: given a WAV format file (assume the total playback time is T', sampling length is L and sampling rate is r_{s}) and a fixed time period $T(T < T')$, the audio file can be divided into many segments based on T, and the time length of each segment is T. We denote K as the number of segments, and K is equal to $\left[\dfrac{T'}{T}\right] + 1$. For each segment, we have that the sampling length is N and we can get $N = r_{\mathrm{s}}T$.

For the segment $[mN, (m+1)N-1]$ $(m = 0, 1, \cdots, K-1)$, we assume that the sample point in order is h_i, where $i = 0, 1, \cdots, N-1$, especially h_0 represents the sample point at mN, and h_{N-1} represents the sample point at $(m+1)N-1$.

Now let us define a function $f(s_m, s_t)$,

$$f(s_m, s_t) = \frac{2\sum\limits_{i=0}^{N-1} h_i x_i}{\sum\limits_{i=0}^{N-1}(h_i^2 + x_i^2)},$$
$$m = 0, 1, \cdots, K-1,$$
$$t = 0, 1, \cdots, (K-1)N,$$

where, s_m and s_t represent the parts $[mN, (m+1)N-1]$ and $[t, t+N-1]$ in the audio file, respectively; h_i and x_i represent the sample points in the parts $[mN, (m+1)N-1]$ and $[t, t+N-1]$ of the audio file, respectively.

The value of this function is the similarity of s_m and s_t segments in the audio file. From the property of the function, it is obvious that the more similar the two segments are, the larger the value is. In addition, when these two segments are totally the same, the value of this function gets to the maximum, that is $f(s_m, s_t) = 1$. On the contrary, if the signal of these two segments is definitely different from each other, the value of the function will become very small, and the value will be close to 0. So we can determine whether the two segments in the audio file are duplicated from the value of the function. For example, if $f(s_m, s_t) \to 1$, the duplication degree is high between the parts $[mN, (m+1)N-1]$ and $[t, t+N-1]$. Instead, if $f(s_m, s_t) \to 0$, the duplication degree is low and the two segments have very different signals.

As a standard for inspecting whether two parts are duplicated, the value of the function is the similarity we defined, and it represents the similarity degree between the parts $[mN, (m+1)N-1]$ and $[t, t+N-1]$ in the audio file.

1.2 Calculation of Similarity

From the defined function above, we can get the multiplication which needs to be carried out: $O(K^2N^2)$, if the function is calculated directly. Obviously, the copulation load is tremendous. It will make a huge discount to the applicability of this method. So it is important to reduce the computation load as lower as possible.

Considering the formula of discrete-time convolution, we know that the value of the function above can be calculated by the convolution:

$$y(k) = h(k) * x(k) = \sum_i h_i x_{k-i}.$$

To calculate $\sum_{i=0}^{N-1} h_i x_i$, we should firstly reverse the data series x_i, and let's call it x'_i. So we can get

$$y(k) = h(k) * x'(k) = \sum_i h_i x'_{k-i} = \sum_i h_i x_i.$$

According to the definition of similarity, it can be divided into three convolutions. The computation of these convolutions decides the whole calculated amount of this procedure. So how to reduce the amount of calculation is the key to improve efficiency.

When we implement the algorithm, the calculating time can be reduced obviously by using Fourier transformation. The experimental results prove that the fast Fourier transformation shortens almost half of the operation time.

Compared with calculation directly, computation load is reduced a lot. In this way these three parts of function can be calculated in order. Then we can obtain the corresponding similarity.

2 Description About the Detect Process

The work flow of detecting duplicated audio segment mainly includes three steps.

(1) We divide the audio file into many segments with the time span of T. This is the essential preparing for the research of next part. And for this division, choosing a proper time span T is significant for us. If the value of T is too small, it will increase the computation load and it has no practical significance. If the value of T is too large, not all duplicate audio inserting parts can be detected, and it will impact the accuracy of our outcome. So under normal circumstances, 0.2 s will be chosen as a duplicate part which is no less than 0.4 s. Thus we can be sure that we never miss hitting a duplicated segment and reduce the computation load.

(2) We need to calculate the similarity which is the standard of the similar degree of two parts in an audio file. The step is the most essential step in our method. In the part 1, we have introduced the definition and meaning of similarity and the fast convolution algorithms to reduce the calculated amount. Besides, the sample rate will have a great impact on the calculation of similarity too. Normally, the sample rate of a WAV format audio is 44.1 kHz; if being used directly, the computation of this program is still huge and will lead to a decrease to the computation efficiency. So computing costs can be reduced and efficiency can be advanced further by reducing the sample rate to an appropriate degree. That is, if the sample rate is reduced about D times, the calculation of the whole program will be decreased by D times. This will improve operating efficiency greatly. At the same time, the reduced multiples can be set arbitrarily according to the accuracy being desired. For convenience, we assume $D = 1$ in this study.

(3) We need to set a threshold to decide which parts are duplicated. If the value of a similarity is larger than this threshold, the relative parts of this similarity are duplicated. From this method, we can know the exact location of duplicate parts too. This threshold can be set based on the precision we want.

The whole process of detecting duplicate audio inserting is shown in Fig. 1. As shown in Fig. 1, we divide the whole audio file into segments whose length is T. We set an initial value from 0 to m, s_m is set as the $(m + 1)$th segment and s_t is set as segment which begins at the $(t + 1)$th sample point in this file. According to the similarity defined above, the similarity value between two segments of the audio file is calculated. Meanwhile, a threshold q is defined to distinguish whether two segments are duplicated; that is, if $f(s_m, s_t) \geqslant q$, they are considered to be duplicated, and they output the start time and end time of relative parts.

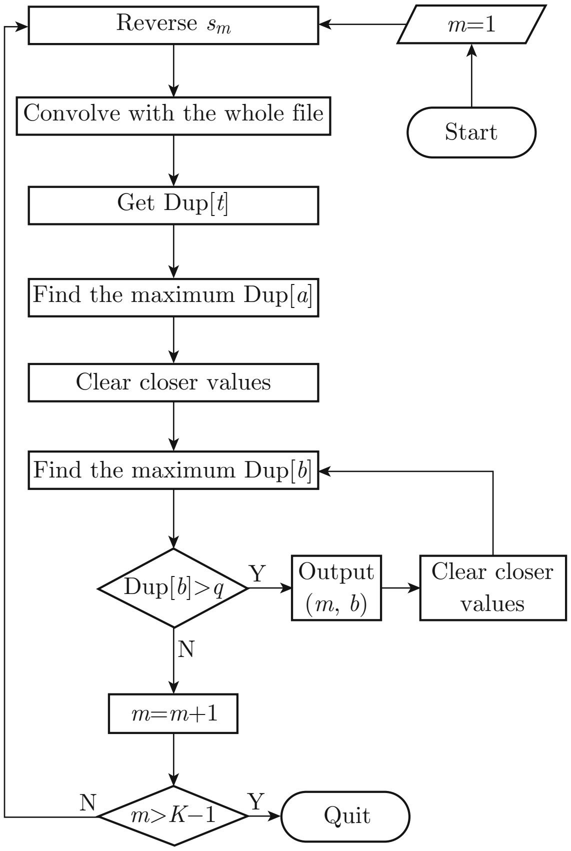

Fig. 1 Description about the detect process

After calculating the convolution sum of s_m and the whole file, we can get the array of $f(s_m, s_t)$ $(t = 0, 1, \cdots, (K-1)N)$. Let's call this array Dup[t]. When we find the maximum value Dup[a] = 1, it's obvious that a equals mN because of $f(s_m, s_t) = 1$. Next, we clear the values beside Dup[a], $\left(\text{Dup}[x] = 0, \text{ for } x \in \left[\max\left\{0, a - \left[\frac{N}{4}\right]\right\}, \min\left\{a + \left[\frac{N}{4}\right], KN\right\}\right]\right)$. Then we search for the maximum value of Dup[t] again, and compare the maximum value Dup[b] with q. If Dup[b] $< q$, we can know that there is no duplication of segment s_m. If Dup[b] $> q$, we should record b and clear the values beside Dup[b], and repeat the operation of search and comparison to find out all the duplicated pieces of s_m.

Next, let $m = m+1$ and implement similar operation to the above one, until $m = K$. By this time every part has compared with the whole audio file. We can get all the position of duplicated segments in the file.

According to the output, we can draw a conclusion whether there is duplicated segment and which segment is duplicated accurately. At the same time we need to know that this conclusion has a great relevant to the threshold value we set, so this threshold should be set cautiously according to the precision we want.

3 Experimental Results

In this section, we will show the effect of checking the duplicated segment by using the method mentioned above. We will design three experiments to check the effect of this method. The first experiment is using a WAV format file with duplicated segments we inserted, while the second experiment without duplication. The last experiment is the research on the impact of audio compression.

3.1 Outcome of a WAV File with Duplication

As the time period (we used) is 0.2 s, the duplicate parts (less than 0.2 s) are overlooked. The outcome of the WAV file with duplicate parts is shown in Fig. 2.

In Fig. 2, we can see that the marks have clearly pointed out the duplicate parts. The next two figures show the similarity of two different segments. One is the duplicated segment, and the other is a non-duplication segment.

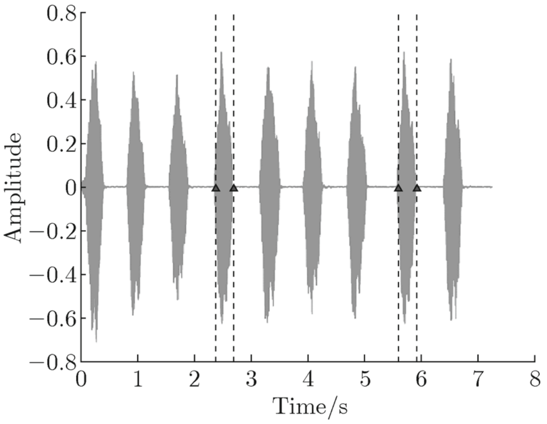

Fig. 2 Outcome of a WAV file with duplicate parts

In Fig. 3, we can see that the duplication degree of several parts in this audio is above the threshold line, so we can get the conclusion that there are duplicated audio segments inserted in the audio file. For contrast, the duplication degree of a segment without duplication is shown in Fig. 4.

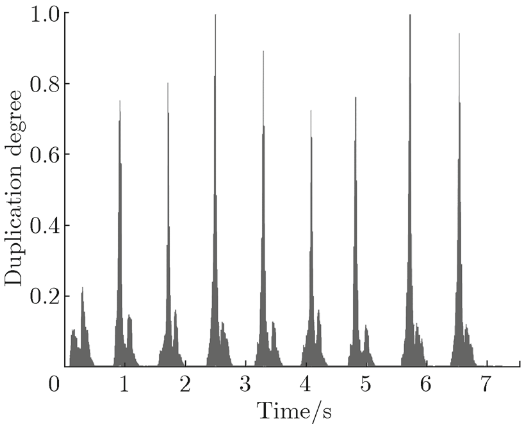

Fig. 3 Duplication degree of a duplicated segment

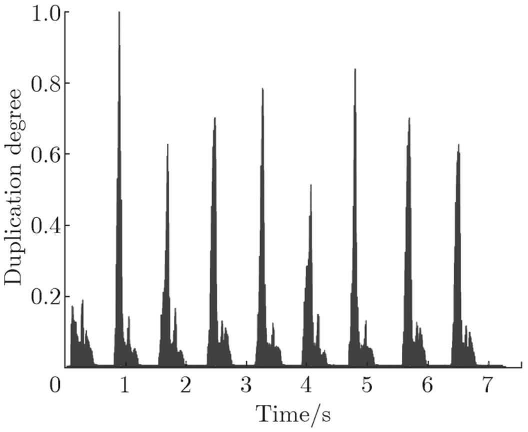

Fig. 4 Duplication degree of a segment without duplication

The outputs of detection coincide with the fact. This method is effective to detect the audio file with duplication.

3.2 Outcome of a WAV File Without Duplication

For the WAV file without duplication, the output is shown in Fig. 5. From Fig. 5, we can see that there is no duplication in the audio file since there is no duplication mark. That is, any segment in the audio file has a low similarity with the rest segments in the file. So we can get a judgment that the audio file has no duplication.

3.3 Impact of Compression

Furthermore, we convert the sample WAV file into MP3 audio file with different compression ratios. And

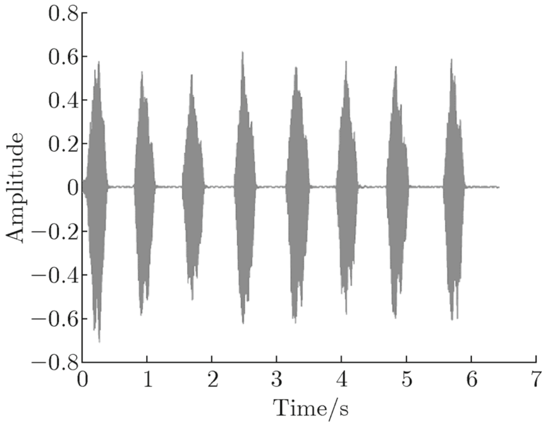

Fig. 5 Outcome of a WAV file without duplication

we design a comparative experiment to find out whether the compression will affect our outcome. The experiment tests the outcome of these different compressed files with different duplication threshold values. If the duplication degree between two segments is larger than the duplication threshold, we will mark it as a duplicated segment.

Table 1 shows our conclusion of the comparative experiment. The tick means that the output is the same as the fact. The cross means that the output has too much marked segments and in fact some of the segments are not duplicated. The triangle means that we fail to mark all the duplicated segments. With the duplication threshold value from 0.95 to 0.98, the compression will not affect the outcome effectively. With a high duplication threshold value like 0.99, the outcome will be affected in case the file is highly compressed. So, as long as we set a proper duplication threshold, we can get outputs which are the same as the fact, no matter whether the audio file has compression or not.

The outcome of a compressed WAV file with duplication is shown in Fig. 6. Compared with the output in the first experiment (without compression), there isn't any obvious difference between the two outputs.

Table 1 Impact of compression with different duplication threshold values

File format	Compression ratio/ (kb· s^{-1})	Duplication threshold						
		0.93	0.94	0.95	0.96	0.97	0.98	0.99
MP3	65	×	×	√	√	√	√	△
MP3	85	×	×	√	√	√	√	△
MP3	100	×	×	√	√	√	√	△
MP3	115	×	×	√	√	√	√	√
MP3	130	×	×	√	√	√	√	√
MP3	165	×	×	√	√	√	√	√
MP3	175	×	×	√	√	√	√	√
MP3	190	×	×	√	√	√	√	√
MP3	225	×	×	√	√	√	√	√
MP3	245	×	×	√	√	√	√	√
WAV	406	×	×	√	√	√	√	√

×—False alarm; √—Correct; △—Omission

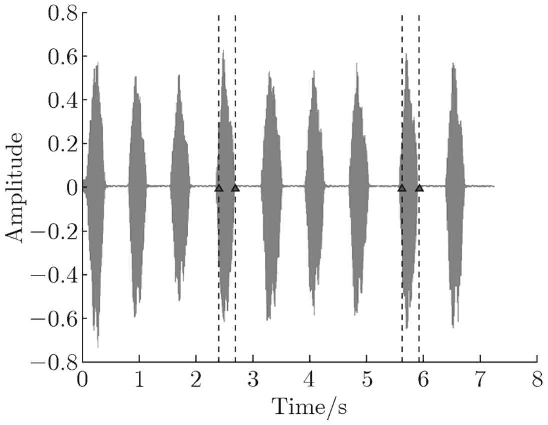

Fig. 6 Outcome of a compressed WAV file with duplication

Through the first and the second experiments, we can know that the outputs are the same as the fact, no matter whether the audio file has duplication or not. And the information about duplicate audio is clear in the outputs, the duplicate information can be got quickly. In the third experiment, we show that our method can also be used for the compressed audio file format, like MP3. We verify the accuracy of this method. It has an important practical significance and application value.

4 Conclusion

This paper proposes a method to detect duplicate audio in a WAV format file. Based on the stability of audio wave, we can make conclusion by comparing audio wave between different segments. In this process, we use the fast convolution to speed up computation. That is, for detecting whether it has duplicated segments in an audio file, this paper proposes the following method. Firstly, we get the sampling data (the sampling rate of the audio file), and make a division to this audio file as a necessary preparation to process. Secondly, we get each segment's separate array which contains similarity values by convolving each segment with the whole audio wave of the file. At last, we compare the similarity values with the threshold value we set, and output the related segments whose similarity value is greater than the threshold value. Then we can come to a conclusion. From the experiments above, we can see that the method we propose has good performance on detecting duplicated segment inserted in WAV file and locating the position of duplicated segments. Though the method we propose is a relatively efficient way, there is still huge room for improvement. Like the problem about reverberation, we know that the reverberation will change the audio wave obviously; however, the reverberation software is not difficult to get. If the counterfeiters reverberate the audio, the result will have some difference with the original one. So it still needs to be improved.

References

[1] FARID H. Detecting digital forgeries using bispectral analysis [R]. Cambridge, USA: Perceptual Science Group, MIT, 1999.

[2] CANO P, BATLE E, KALKER T, et al. A review of algorithms for audio fingerprinting [C]//*Proceedings of 2002 IEEE Workshop on Multimedia Signal Processing*. Piscataway, USA: IEEE, 2002: 169-173.

[3] GRIGORAS C. Digital audio recording analysis: The electric network frequency criterion [J]. *International Journal of Speech Language and the Law*, 2005, **12**(1): 63-76.

[4] SINITSYN A. Duplicate song detection using audio fingerprinting for consumer electronics devices [C]//*Proceedings of 2006 IEEE Tenth International Symposium on Consumer Electronics* (*ISCE'06*). Piscataway, USA: IEEE, 2006: 1-6.

[5] YAO Qiu-ming, CHAI Pei-qi, XUAN Guo-rong, et al. Audio re-samplingdetection in audio forensics based on EM algorithm [J]. *Computer Applications*, 2006, **26**(11): 2598-2601(in Chinese).

[6] KRAETZER C, OERMANN A, DITTMANN J, et al. Digital audio forensics: A first practical evaluation on microphone and environment classification [C]//*Proceedings of the 9th Workshop on Multimedia and Security*. New York, USA: ACM, 2007: 63-74.

[7] YANG R, QU Z, HUANG J. Detecting digital audio forgeries by checking frame offsets [C]//*Proceedings of the 10th ACM Workshop on Multimedia and Security*. New York, USA: ACM, 2008: 21-26.

[8] MAHER R C. Audio forensic examination: Authenticity, enhancement, and interpretation [J]. *IEEE Signal Processing Magazine*, 2009, **26**(2): 84-94.

[9] MAHER R C. Overview of audio forensics [C]//*Intelligent Multimedia Analysis for Security Applications*. Berlin, Germany: Springer-Verlag, 2010: 127-144.

[10] RODRÍGUEZ D, APOLINÁRIO J, BISCAINHO L. Audio authenticity: Detecting ENF discontinuity with high precision phase analysis [J]. *IEEE Transactions on Information Forensics and Security*, 2010, **5**(3): 534-543.