

One approach in the time domain in detecting copy-move of speech recordings with the similar magnitude

Nguyen Tuan Anh, Han Thi Thuy Hang, Guang Chen

Abstract— In this study, we focused on copy-move detection in the content of audio files. We have many ways to identify a forged audio, in this article we focus on comparing the pitch of samples, compare the similarity of sound intensity in the time domain together. The goal of this method is to focus detecting copy-move speech in an audio file, which it has archived high speed processing and improved accuracy.

Index Terms— audio forensics, audio forgeries, magnitude similarity, copy-move detection

I. INTRODUCTION

Today the digital audio files play an important role in evidence at the trial. Unfortunately, there are numerous audio editing software, it is easy to use and dissemination, they can easily change the content of an audio track, especially changing the content of the speech.

Imagine that, in a trial, prisoner have declared and pleaded guilty, because of cheating, so people edit audio files making him not guilty, by copying and pasting content in a file, the listener can not identify an audio file that is accurate or not, because the entire speech in the audio file is of the inmates. Audio files can be filtered, added noise, to create forged audio files.

In recent years, there are many methods to identify forged audio files, such ENF, recompress detection, noise environment and bispectral analysis. There are many articles on the identification of forged based audio on copy-move approach[8]-[12]. However, different approaches bring different results. Here, I provide a method comparable similarity threshold peaks in the time domain. The method I propose, just concentrate on the sounds in meaningful areas, remove the silence, quantized samples, comparing only the peak level when the length of the voices are very similar. Because we have removed unnecessary segments, there are no redundant processing components, it can increase processing speed.

II. PROPOSED METHOD

Identify magnitude and magnitude processing. Each audio file to use as evidence in court are often long, to detect sample copy paste each one takes a lot of time. After several digital signal processing, it is also more difficult to detect. Therefore, the problem to be achieved here is short-term detection, low

Nguyen Tuan Anh, School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, P.R.China, Phone/ Mobile No. (+86)13250282662

Han Thi Thuy Hang, Faculty of Civil and Environment, Thai Nguyen University of Technology, Thai Nguyen, Vietnam

Guang Chen, 9-11 Kelin road, High-tech Industrial Development Zone, Guangzhou 510640, P.R.China

number of operations and it can obtain good results after the file is processed. We found the method that compares the magnitude of the voice for each segment.

A. Basic knowledge of Magnitude

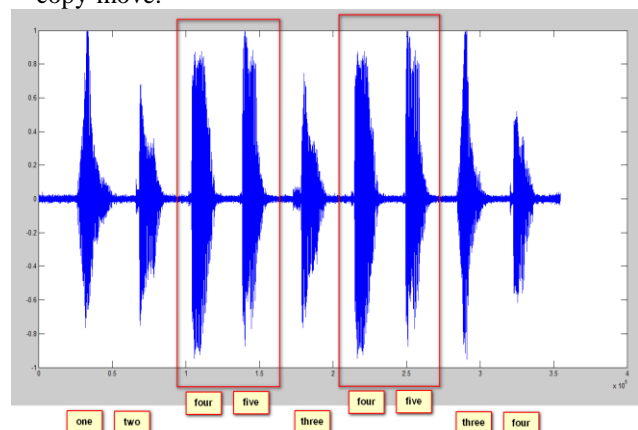
We consider a basic example, for an audio that has read the numbers, through audio processing software, we copy some of the clips and paste them in another place to create a forged audio file. The magnitude of the voice is a set of amplitudes at all times represented in the time domain. We can easily see the amplitude of each sound through speech processing software. Each sound is played by the same speaker, with one syllable, but often not the same samples by samples. Especially when analyzing the differences between the magnitudes, we can see very clear differences. According to Figure 1 for example, in one sentence we have the word "one", the 2nd is copied from the first word, magnitudes of the first and the second word is relatively similar and different from the magnitude of the 3rd.

B. Method of extracting magnitudes

Extract magnitudes is a classic method of digital signal processing. There are many traditional methods to extract magnitudes, such as Average Magnitude Difference Function (AMDF), autocorrelation Functions (ACF), and Cepstral. But the traditional method to extract magnitudes has a few disadvantages, So I proposed extraction method based on time domain.

The steps to take are as follows:

- 1) Negated all negative values in the audio file, select the threshold for removing noise of the entire file. All values are less than the threshold, to be 0.
- 2) Quantized under the window, the window width is w
- 3) Count the length of each word.
- 4) Comparing amplitude of the nearly equal length, the smallest difference between words that have the copy-move.



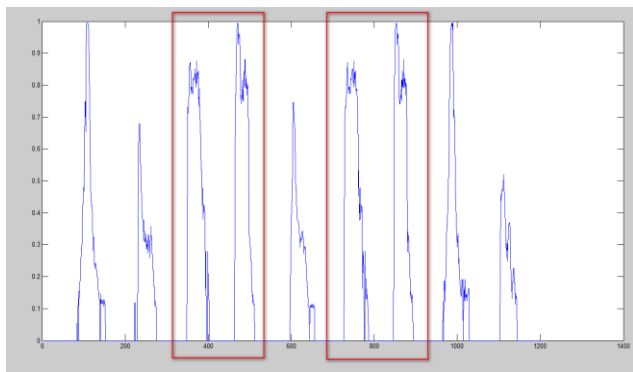


Figure 1. Analysis of magnitudes, original audio and Quantized audio

III. IMPLEMENTATION

The process is divided into 2 parts, extract magnitudes and comparison between the magnitudes of word. It is illustrated in Figure 2.

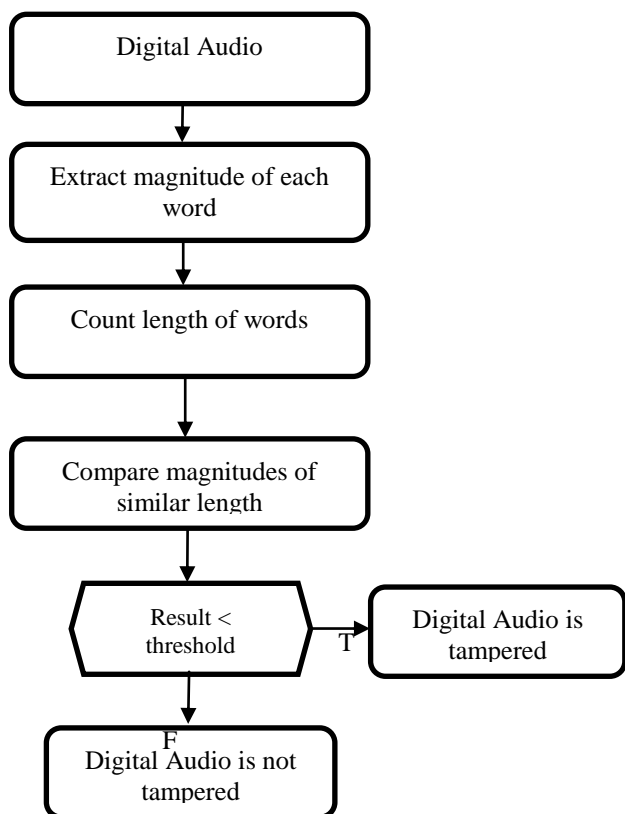


Figure 2. Summary of detection process

Magnitude extraction method.

In Section 2, we introduce the Magnitude and methods to extract it. in this section we will explain more clearly about processing method.

First we reverse all negative values.

$$x(x < 0) = x(x < 0) * (-1) \tag{1}$$

Where x is the time domain value of speed signal.

Determine the threshold to filter noise, noise filtering threshold selected here by 0.15

$$x(x \leq 0.15) = 0 \tag{2}$$

Quantized.

$$M(t) = \text{Max}_t^{n*(t+1)}(x) \tag{3}$$

Where t is the samples of quantized sequences, n is the frame length of window.

Magnitude sequences comparison.

we have the sequence magnitude for each word, now we measure the length of each word, if the length nearly equal, compare them with each other. There are many methods to compare the two series, here I use the method of the smallest deviations (SD). SD is defined as follows:

$$SD = \sum_{k=1}^{N_{min}} |x(k) - y(k)| \tag{4}$$

Where Nmin is shorter than the length of the sequence, x (n) and y (n) is two words, the words are the same, then the smaller SD result. In Figure 3, we have the result of forged audio files, the results marked as words, it has been copied to create forged, discovered by the algorithm.

	1	2	3	4	5
1	230	597	45	46	4.2261
2	230	967	45	46	12.0387
3	230	348	45	47	15.8488
4	230	464	45	47	14.8832
5	230	846	45	47	14.5415
6	597	967	46	46	10.8658
7	597	348	46	47	14.5224
8	597	464	46	47	13.7004
9	597	846	46	47	13.2039
10	597	727	46	48	14.4493
11	967	348	46	47	12.4257
12	967	464	46	47	12.4434
13	967	846	46	47	11.9151
14	967	727	46	48	12.2866
15	348	464	47	47	5.5760
16	348	846	47	47	5.5278
17	348	727	47	48	0.3799
18	464	846	47	47	0.7664
19	464	727	47	48	5.2943
20	846	727	47	48	5.2461
21					
22	Pos1	Pos2	L1	L2	result
23					
24					

Where Pos1, Pos2 are positions of tampered words in Quantized data, L1, L2 are word lengths at Pos1, Pos2 positions, respectively. Result is the calculated by SD equation, forged mean low value.

In figure 3, we have 20 pairs of approximately the same length, after comparing the results, it has two pairs of the same, since, seen in this file, there are two words have been copy-move. The values are very small in comparing with others. This time, we only need chose the second threshold value to auto determine tampered words

IV. EXPERIMENTAL RESULTS

Data. We prepare 10 audio data original files without data forged duplicate copy-move, the timing of all audio files from 1:29s to 21:26 seconds, these files were tempered by copy then paste in to themselves, and received 486 post-processing forged files. The data sets used in the experiment consists of these audio files.

With this database, the percent of correct value (PC) detection was calculate by formula 5

$$PC = \frac{C}{W + C} * 100\% \quad (5)$$

Where C is number of correct result, W is the number of wrong results.

With this database, this method produced PC=91.38%, showed as in Figure 4.

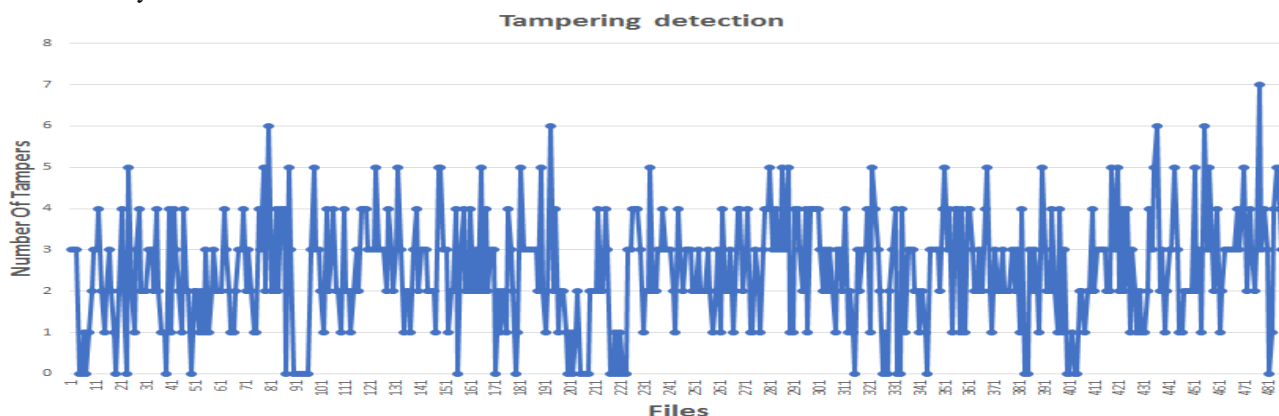


Figure 4. SD of tampered speech audio

Time calculations. We prepared 35s audio file, the sampling frequency is 44 kHz, mono. Processing time is 1.9122s for the program giving the final result, while we experiment with the extracted magnitudes yaapt [1] (a fundamental frequency (the magnitude) tracking Algorithm), extract magnitudes time of yaapt is 44.6467s.

V. CONCLUSION

Derived from the simplest and most common feature in Audio modification, that is copy and paste. This article offers an improved version of the copy-move forged identification method, simplified formulas, increased accuracy and reduced processing time. In the coming time, the author will invest more time in forgery audio identification with shorter forged content.

REFERECES

- [1]. YAAPT (Yet Another Algorithm for Magnitude Tracking), ws2.binghamton.edu/zahorian/yaapt.htm, 2016-06-06.
- [2]. Copy-move detection of audio recording with magnitude similarity, Qi Yan, School of Information Science and Technology, Sun Yat-sen University, China, 2015 IEEE
- [3]. B. D'Alessandro, Y. Shi, *MP3 Bit Rate Quality Detection through Frequency Spectrum Analysis*, International Multimedia Conference, Proceedings of the 11th ACM workshop on Multimedia and security Princeton, 2009.
- [4]. C.Grigoras , *Digital Audio Recording Analysis: The Electric Network Frequency (ENF) Criterion*, The International Journal of Speech Language and the Law , 12(1), pp:63-76 , 2005.
- [5]. R. Garg, A. Varna, and M. Wu. "Seeing ENF: natural time stamp for digital video via optical sensing and signal processing." Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011.
- [6]. H. Farid, *Detecting Digital Forgeries Using Bispectral Analysis*, MIT AI Memo AIM-1657, MIT , 1999.
- [7]. J. Fridrich, B. Soukal, and A. Lukáš. "Detection of copy-move forgery in digital images." in Proceedings of Digital Forensic Research Workshop, 2003.

- [8]. Z. Xie, W. Lu, X. Liu, Y. Xue, and Y. Yeung, "Copy-move detection of digital audio based on multi-feature decision," J. Inf. Secur. Appl., vol. 43, pp. 37–46, 2018.
- [9]. R. Susan Oommen and ayamohan M, "A Hybrid Copy-Move Forgery Detection Technique Using Regional Similarity Indices," Int. J. Comput. Sci. Inf. Technol., vol. 7, no. 4, pp. 157–134, 2015.
- [10]. M. Imran, Z. Ali, S. T. Bakhsh, and S. Akram, "Blind Detection of Copy-Move Forgery in Digital Audio Forensics," IEEE Access, vol. 5, pp. 12843–12855, 2017.
- [11]. A. B. Patankar, A. Sukhpal, S. Shetye, and A. ShroffShroff, "Copy move forgery detection," 2017, pp. 6 -6 .
- [12]. M. Qiao, A. Sung, Q. Liu, and B. Ribeiro, "A novel approach for detection of copy-move forgery," in proceedings of Fifth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2011), 2011, pp. 44–47.