Article

Exposing Speech Resampling Manipulation by Local Texture Analysis on Spectrogram Images

Yujin Zhang ^{1,2,*} , Shuxian Dai ¹, Wanqing Song ¹, Lijun Zhang ¹ and Dongmei Li ¹

¹ School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; dai_shuxian@163.com (S.D.); song_wanqing@sues.edu.cn (W.S.); zhang_lijun@163.com (L.Z.); dmlis@sues.edu.cn (D.L.)

² Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai Jiao Tong University, Shanghai 200240, China

* Correspondence: yjzhang@sues.edu.cn; Tel.: +86-1521-671-2378

Received: 23 November 2019; Accepted: 16 December 2019; Published: 25 December 2019

Abstract: Speech tampering may be aided by the resampling operation. It is significant for speech forensics to effectively detect the resampling; however, there are few studies on speech resampling detection. The purpose of this paper was therefore to provide a new training ideal to detect speech resampling. After resampling, the speech signal changes regularly in the time–frequency domain. In this paper, we theoretically analyzed the corresponding relationship between time domain and frequency domain of the resampled speech. Compared with the original speech, the bandwidth of resampled speech was stretched or compressed. First, the spectrogram was generated by short-time Fourier transform (STFT) from the speech. Then, the local binary pattern (LBP) operator was applied to model the statistical changes in the spectrogram and the LBP histogram was calculated as discriminative features. Finally, a support vector machine (SVM) was applied to classify the developed features to identify whether the speech had undergone the resampling operation. The experimental results show that the proposed method has superior detection performance in different resampling scenarios than some existing methods, and the proposed features are very robust against the commonly used compression post-processing operation. This highlights the promising potential of the proposed method as a speech resampling detection tool in practical forensics applications.

Keywords: speech resampling detection; spectrogram; texture analysis; local binary pattern

1. Introduction

The digital information age has accelerated the development of multimedia, which is now universally present in people's daily lives. Common multimedia includes video, image, audio, and so on. Due to the popularity and intelligence of media editing tools, people can edit and spread media easily, making the primitiveness and authenticity of multimedia unguaranteed. Multimedia forensics technology has become a leading research topic in the field of information security. Multimedia forensics can be divided into active forensics and passive forensics. Active forensics requires embedding prior information (called a watermark) into the media before the media is released. The extracted watermark is compared with the original watermark at the receiving end, and the primitiveness and authenticity of the media are identified according to the changes in the watermark. On the contrary, passive forensics is a direct detection of media information without prior information. The application scenario of passive forensics is more extensive, but its technical requirements are also higher. Recently, passive image forensics has been the subject of many state-of-the-art studies [1–4], but the research on passive audio forensics is still in its infancy [5]. Forensic scenarios such as speech source detection [6], speech steganalysis [7], and speech recapture [8], etc., are the categories of passive speech forensics.

Speech tampering tricks include copy-paste, splicing, recompression, recapture, and so on. In recent years, the detection of copy-paste is receiving more attention in speech forensics, which can be detected by principal component analysis (PCA) [6], discrete Fourier transform (DFT) [9], and other algorithms. Speech recompression can be detected by frame inconsistency [10]. According to the high-order correlation in frequency domain, speech recapture identification can achieve promising results [8]. In the field of digital signal processing, the conversion of sampling rate is often encountered. Some forging tricks such as splicing, electronic disguising, and quality faking require the adjustment of the sampling rate to create a realistic speech. It is difficult for the human ear to distinguish whether an audio is resampled or not. Therefore, the detection of speech resampling can be a good assistant for speech forensics.

So far, the detection of speech resampling is still a challenge. Most of the existing detection methods of speech resampling are inspired by image resampling detection methods. In some existing work, speech resampling is detected by analyzing interpolation. Yao et al. [11] analyzed the discrete peak value of speech in frequency to detect speech resampling under the assumption that resampling would introduce periodic artifacts to original speech. However, such a method is only effective for linear interpolation. Chen et al. [12] improved the performance in the cases of different interpolation kernels. The methods [11,12] are greatly influenced by the initial values of parameters and have unsatisfactory detection rates for down-sampling. To avoid this problem, Shi and Ma [13] analyzed the resampling interpolation and detected it by researching the varieties in the linear relationship between the sampling points. After the speech is analyzed by singular value decomposition, a statistical zero singular value average is generated to describe the statistical change, which avoids the problem of high complexity of expectation maximization (EM) algorithms. Unlike analyzing interpolation, Qi and Peng [14] put forward the concept of sub-band smoothing. Based on the fact that the resampled signal loses part of the high-frequency information after low-pass filtering, they computed the smoothness of different sub-bands to detect the speech resampling. However, linear interpolation and nearest-neighbor interpolation have a larger smoothness in the high-frequency part, which affects the detection performance of the method. Inspired by the periodic neighborhood correlation of the image after resampling, Hou et al. [15] analyzed the periodicity of the second-order difference of the speech signal and used the absolute value to replace the variance of second-order difference for automatic detection. However, this method needs to set a threshold to search for the peak value, which is greatly influenced by different interpolation methods. Wang et al. [16] proposed a resampling detection method based on inconsistency between bandwidth and sampling ratio from the perspective of speech characteristics, using the receiver operating characteristic curve to find the best threshold to judge speech resampling based on the logarithmic energy ratio. This method does not work well in the case of speech down-sampling.

Based on the existing research, this paper started with the relationship between the period and frequency of speech in the resampling process, and used a local binary pattern (LBP) operator to extract the features from the spectrogram. Further, it used the statistical histogram of LBPs to distinguish the resampled speech from the original one by using a support vector machine (SVM). Experimental results show that the proposed method can detect speech resampling effectively under different sample factors and compression bit rates.

The rest of this article is arranged as follows. In Section 2, the principle of resampling is briefly introduced. In Section 3, the relationship between resampling in time domain and frequency domain is studied theoretically. In Section 4, this paper introduces the proposed feature extraction method. In Section 5, the experimental results of the resampling detection under the different interpolation conditions are considered, and the experimental comparison for different compression bit rates are also given. Finally, in Section 6, conclusions are drawn and future work directions are pointed out.

2. Principle of Resampling

For this paper, the characteristics of signal resampling were taken as the starting point, in order to detect the tampering features. In this section, the principle of resampling is analyzed, which leads to the rule of change on the spectrogram image. The change in the signal's integer multiple only involves up-sampling, interpolation, or down-sampling, and changes in any multiple requires the combination of up-sampling, interpolation, and down-sampling. Suppose the original signal $x(n)$ has m sampling points, p is the interpolation factor, q is the decimation factor, and the resampling factor is expressed as p/q :

(1) Up-sampling: $p - 1$ zero points are inserted between every two points of the signal, and the sampling point of the changed signal $x_u(n)$ becomes $p(m - 1) + 1$, which enlarges the signal $x(n)$ by a factor of p in the time domain. When $n = 1, 2, \dots, m$, $x_u(p(n - 1) + 1) = x(n)$; otherwise, $x_u(n) = 0$.

(2) Interpolation: By convoluting the signal $x_u(n)$ with the low-pass linear interpolation filter $h(n)$, the interpolated signal $x_c(n) = x_u(n) * h(n)$ can be obtained, which can be divided into linear interpolation, spline interpolation, and cubic interpolation according to the different filters [17].

(3) Down-sampling: Take one point for each q point, that is, the interval between two points is $q - 1$, and this reduces the signal $x_c(n)$ by a factor of q in the time domain. The changed signal is $x_d(n) = x_c(1 + q(n - 1))$; when $n = 1, 2, \dots, \frac{(p-1)m}{p}, \frac{(p-1)m}{p} + 1$, the resampled signal is $y(n) = x_d(n)$.

Therefore, if $p > q$, there is an increase in the signal in the time domain, whereas if $p < q$, there is a decrease in the signal in the time domain. Different types of resampling algorithms can be divided into the above three steps. The difference between speech resampling methods lies in the interpolation filters used in step (2). In this paper, we also discuss the influence of several common interpolation methods on the spectrogram.

3. Effect on the Spectrogram of Speech Resampling

3.1. Spectrogram

It is well known that speech is a nonstationary signal, so the premise of generating a spectrogram is that the speech can be considered stable for a certain period of time. First, we framed the speech signal $x(n)$. The frame length used in this paper was set to 50 ms and the value of the frame shift was 25 ms. The Fourier transform was performed on each frame, and the Fourier transform form $X(k, t)$ related to the t -th frame is given by the following:

$$X(k, t) = \sum_{n=0}^{d-1} x(n)w(n)e^{-\frac{2kn\pi}{d}}, k = 0, \dots, d - 1, \quad (1)$$

where d is the frame length and $w(n)$ is the Hamming window function. Each frame is arranged in a frame shift time to form a linear spectrogram $L(k, t)$, which can be formulated as follows:

$$L(k, t) = |X(k, t)|, \quad (2)$$

where $|\cdot|$ stands for the amplitude operator and $t \in \{1, \dots, \frac{n-25}{50-25}\}$. A logarithmic spectrogram $S(k, t)$ can be generated by taking the logarithm of the linear spectrogram [18], which can be expressed as follows:

$$S(k, t) = \log(L(k, t)). \quad (3)$$

The purpose of converting a linear spectrogram to a logarithmic one is to reduce the dynamic range of amplitude components by using logarithmic compression. In fact, what the spectrogram presents is three-dimensional information. The horizontal axis represents time, that is, the length of speech; the vertical axis represents frequency; and the depth of color represents the frequency

amplitude [19]. Pitch period, frequency band width, and fundamental frequency can be extracted from the spectrogram.

3.2. Relationship between Resampling and Spectrogram

As can be seen from Section 3.1, Fourier transform is a crucial part of the spectrogram generation, so the traces introduced by speech resampling will also be reflected on the spectrogram. Speech can be regarded as a complex signal consisting of k -order harmonics. This section analyzes the relationship between the time domain and frequency domain of k -order harmonics to illustrate the relevant phenomena after the resampling manipulation on the spectrogram.

For harmonic periodic function $x(t)$, for example:

$$x(t) = A \cos(\omega t + \varphi), \quad (4)$$

where A is the magnitude, φ is the phase angle, and $\omega/(2\pi)$ is the frequency of the function, which can be written using trigonometric equations:

$$x(t) = A \cos \varphi \times \cos \omega t - A \sin \varphi \times \sin \omega t. \quad (5)$$

Since φ is a constant, let $a = A \cos \varphi$, $b = -A \sin \varphi$; it can be simplified as follows:

$$x(t) = a \cos \omega t + b \sin \omega t. \quad (6)$$

The Fourier expansion $f(n)$ of the signal is as follows:

$$f(n) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(\frac{k\pi n}{l}) + b_k \sin(\frac{k\pi n}{l})). \quad (7)$$

Sampling must be introduced between the processes where signals convert analog signals into digital signals. This processes that for the sampling interval for Δt , k is the order of harmonic, l is one half of the period, N is the sampling number, and n corresponds to the time sequence of time, so the $n \rightarrow i\Delta t$, therefore:

$$\frac{k\pi n}{l} \rightarrow \frac{k\pi i\Delta t}{N\Delta t/2} = \frac{2k\pi i}{N}. \quad (8)$$

Applying it to the above equation, the discrete Fourier series can be obtained as follows:

$$n_i = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos(\frac{2k\pi i}{N}) + b_k \sin(\frac{2k\pi i}{N})) \quad (9)$$

In the equations:

$$a_0 = (\frac{2}{N}) \sum_{i=1}^{N-1} x_i, a_k = (\frac{2}{N}) \sum_{i=0}^{N-1} x_i \cos(\frac{2k\pi i}{N}), \quad (10)$$

$$b_k = (\frac{2}{N}) \sum_{i=0}^{N-1} x_i \sin(\frac{2k\pi i}{N}), k = 1, 2, \dots, m, \quad (11)$$

where $k = 0$, a_0 is the DC component, and the k -order harmonic of x_i can be obtained as follows:

$$a_k \cos(\frac{2k\pi}{N\Delta t}(i\Delta t)) + b_k \sin(\frac{2k\pi}{N\Delta t}(i\Delta t)). \quad (12)$$

The frequency f_k and period T_k of the k -order harmonic are as follows:

$$f_k = \frac{k}{N\Delta t}, T_k = \frac{1}{f_k} = \frac{T}{k}, \quad (13)$$

where T is the time of the speech taken. It can be observed from the equation that the frequency of each harmonic function is inversely related to the total time length, which also confirms the scaling characteristics in the Fourier transform:

$$F(f(ct)) = \frac{1}{|c|} F\left(\frac{w}{c}\right), (c \neq 0), \quad (14)$$

where f is the signal in the time domain, F represents the Fourier transform of the signal, and c is a constant. Therefore, it can be concluded in this paper that when the time domain period changes in multiples, the range of frequency domain also changes correspondingly. For example, when the speech is up-sampled by the factor p , it is equivalent to stretching p times in the time domain, and the frequency of the harmonics will compress p times. At the time of down-sampling, q is applied to the discrete signal again. As shown in Figure 1, the signal shortens by q times in the time domain and stretches by q times in the frequency domain.

From Figure 1, it can be confirmed that the bandwidth of resampled speech is stretched or compressed in the frequency domain, which can be extended to the spectrogram. With the change in the sampling factor, the period of speech in the time domain changes, the bandwidth on the longitudinal axis changes correspondingly, and each harmonic has the same positional movement.

Figure 1. Discrete Fourier transform (DFT) amplitude spectra of (a) the original speech, (b) the resampled speech with $p/q = 0.5$, and (c) the resampled speech with $p/q = 2$.

Figure 2 shows the influence of different sampling factors and compression bit rates on the spectrogram. In the bottom left corner is the spectrogram of the original speech. The horizontal axis represents the change in the sampling factor, and the vertical axis represents the change in the MP3 compression bit rate. It can be observed from the horizontal axis and the frequency positions of each harmonic that, after the down-sampling process, the corresponding spectrogram is stretched and the characteristics of the spectrogram are changed when compared with the original speech. At the time of speech up-sampling, the spectrogram is compressed correspondingly, which can be observed from the bandwidth of the whole spectrogram. We observe from the vertical axis that different compression bit rates have no negative effect on the spectrogram, but the effect of resampling still exists. Therefore, the spectrogram-based resampling detection method has a better anti-mp3 compression performance.

Figure 2. Spectrogram distribution in different situations.

Figure 3 shows the changes in the spectrogram under different interpolation methods of the resampled speech, and the resampling factors are all 1.5. It can be seen that except for the zero-complementing interpolation, the common three interpolation methods leave energy residue in the blank band of the spectrogram, but the change pattern still exists. In Section 5, we present the experimental comparison for different interpolation cases.

Figure 3. Cont.

Figure 3. Spectrograms under different interpolation methods. (a) Resampled speech generated by zero-complementing interpolation; (b) resampled speech generated by linear interpolation; (c) resampled speech generated by spline interpolation; (d) resampled speech generated by cubic interpolation.

4. Feature Extraction

After the speech is resampled, the spectrogram is compressed or stretched, and the texture features of the image change accordingly. An LBP histogram is a kind of texture description operator, which is used to extract rich local texture information from images [20]. It puts more emphasis on texture features and the variation of elements in adjacent areas, and can extract local features of the spectrogram well, thereby reflecting the variation pattern of the resampled speech. Therefore, this paper used the LBP histogram to model the local texture change in the spectrogram image.

The change information in the arbitrary direction of the spectrogram was then represented by the difference between the eight-neighborhood elements and the center element. The steps to extract the spectral features using an LBP operator are as follows.

First, the spectrogram was divided into 3×3 regions, as shown in Figure 4a. The LBP compares the value of the center element with the surrounding eight elements; the neighborhood element smaller than the center element is marked as 0, otherwise it is marked as 1. The binary value output is calculated using Equation (16). The binary value is multiplied by the weight and then added to obtain the LBP code L_C of the central element [21]. The L_C represents the LBP code of the central element, T is the threshold, S_i stands for the value of each element in the spectrogram, and C_i is the binary value output of the LBP.

$$L_C = \sum_{i=1}^7 C_i \times 2^i, \quad (15)$$

$$C_i = \begin{cases} 0, & S_i - S_c < T \\ 1, & S_i - S_c \geq T \end{cases}, \quad (16)$$

where the value of the threshold T is set to 0.

Figure 4. The value of the local binary pattern (LBP) is $8 + 16 + 32 + 64 = 120$. (a) The 3×3 region; (b) the calculation of the LBP.

Finally, the histogram of these codes is calculated to represent the characteristic image [22]. Figure 5 is the comparison of the LBP histograms computed from an original speech and its resampled versions.

As can be seen from Figure 5, the local texture features extracted from the spectrogram of the original speech are significantly different from those of the resampled speech. The underlying reason is that the spectrogram is stretched or compressed after resampling manipulation, which leads to the changes in the statistical correlation between the spectrogram's elements. Therefore, the local details and local textures are correspondingly changed, which can be depicted effectively by the LBP operator. Furthermore, the obtained LBP histogram feature can be fed into the LIBSVM [23] to distinguish the original speech from the resampled one. The LIBSVM is a software package based on the support vector machine.

Figure 5. Comparison of the LBP histograms. LBP histograms computed from (a) an original speech, (b) the down-resampled speech ($p/q = 0.6$), and (c) the up-resampled speech ($p/q = 2.0$).

For the proposed method in this paper, we consider the maximum complexity of each part as the entire complexity of the algorithm, then the time complexity of the spectrogram is $T(n) = O(n \log_2 n)$ and the LBP is $T(n) = O(n^2)$. Thus, the time complexity of the proposed algorithm can be written as $T(n) = O(n^2)$. The proposed speech resampling detection process is as follows and shown in Figure 6.

- (1) Use short-time Fourier transform to transform input speech into a spectrogram;
- (2) Then divide the spectrogram into 3×3 regions to calculate the LBP code;

- (3) Subsequently, calculate the histogram of these codes for feature representation.
- (4) Finally, the obtained LBP histogram feature can be fed into the LIBSVM to distinguish the original speech from the resampled one.

Figure 6. Proposed speech resampling detection process.

5. Experimental Results and Analysis

The original speech dataset in the experiment was selected from the TIMIT corpus. This corpus contains 6300 utterances, with an average duration of three seconds, from 630 participants in different regions of the United States. The original speech files are mono, PCM encoding, 16 kHz sampling ratio, and 16-bit quantization [16]. In this paper, 3000 original speech samples were selected from the TIMIT corpus. The original speech was resampled by linear, spline, and cubic interpolations separately and the sampling factor was changed from 0.6 to 2 with a step of 0.1 to achieve the resampled databases with linear, splines and cubic interpolations. Then, we used MP3 compression with three different bit rates, namely 32 kbps, 64 kbps, and 128 kbps, to compress the original speech and resampled speech, yielding a compressed speech database with MP3 compression. In order to facilitate the experiment, a mixed corpus was constructed by randomly selecting resampled speech from different interpolation databases (i.e., linear, spline, and cubic interpolations). There was a total of 684,000 speech samples used for the experiment, and the process of making the corpus is shown in Figure 7.

Figure 7. Production process of the corpus.

5.1. Experimental Results under Different Compression Interpolation Methods

In Section 2, this paper introduced three steps of speech resampling, namely up-sampling, interpolation, and down-sampling. There are several interpolation methods in resampling, which mainly include linear interpolation, spline interpolation, and cubic interpolation. Therefore, this paper also discusses the influence of various interpolation methods on the detection accuracy of resampling. When the resampling factors are 0.8, 0.9, 1.1, and 1.2, the speech tampering degree is so small that it is difficult to detect. Consequently, this section uses these four representative sampling factors for a comparative study. The kernel of LIBSVM is a radial basis function (RBF), and we use the multiplicative grid of five-fold cross-validation to select the parameters with the smallest error, and the ranges of the parameters c and g are all from 2^{-5} to 2^5 . It can be concluded from Tables 1–3 and Figure 8 that when the resampling factor is greater or smaller than 1, the detection accuracy of all methods is high. Generally speaking, under different interpolation methods, it can be seen that the detection accuracy of the proposed algorithm is higher than that in Wang’s and Hou’s work. It is indicated that the proposed algorithm significantly outperforms these state-of-the-art studies.

For experimental convenience, except for this section, the following experiments in this paper use the mixed resampling corpus of linear interpolation, spline interpolation, and cubic interpolation for the research.

Table 1. Detection results of the method proposed in this paper under different interpolation methods.

p/q	Mixed Interpolation	Linear Interpolation	Spline Interpolation	Cubic Interpolation
0.8	0.8360	0.8340	0.8270	0.8553
0.9	0.7030	0.6600	0.6950	0.6960
1.1	0.7530	0.7440	0.7430	0.7413
1.2	0.8160	0.8460	0.8000	0.8127

Table 2. Detection results of Wang et al. [16] under different interpolation methods.

p/q	Mixed Interpolation	Linear Interpolation	Spline Interpolation	Cubic Interpolation
0.8	0.4943	0.4820	0.5310	0.4950
0.9	0.4490	0.4140	0.4840	0.4550
1.1	0.6247	0.6300	0.6690	0.6700
1.2	0.7027	0.6620	0.7490	0.7030

Table 3. Detection results of Hou et al. [15] under different interpolation methods.

p/q	Mixed Interpolation	Linear Interpolation	Spline Interpolation	Cubic Interpolation
0.8	0.7061	0.7237	0.7061	0.6841
0.9	0.6876	0.6771	0.6619	0.6673
1.1	0.6902	0.6514	0.6806	0.6817
1.2	0.6657	0.6745	0.6201	0.6675

Figure 8. Influence of different interpolation conditions on resampling detection. (a) The detection accuracy in the case of linear interpolation; (b) the detection accuracy in the case of spline interpolation; (c) the detection accuracy in the case of cubic interpolation; (d) the detection accuracy in the case of mixed interpolation.

5.2. Experimental Results under Different Compression Sampling Factors

In previous work, Wang et al. used the logarithmic ratio of statistical energy to find the best threshold for resampling detection based on the inconsistency between bandwidth and sampling ratio. Hou et al. [15] employed the periodicity of the second-order difference of the speech signal to identify resampling. When the sampling factor is greater than 1.2, these methods can achieve better detection results. In the case of a sampling factor from 0.8 to 1.1, their method cannot achieve the satisfactory performance. Therefore, this paper used the mixed corpus to carry out the detection accuracy in the case of a sampling factor from 0.6 to 2, and the detection accuracy was improved to a certain extent. The experimental results are shown in Table 4 and Figure 9.

Table 4. Mixed resampling detection results of the proposed method and the state-of-the art studies.

p/q	Proposed Method	Wang et al. [16]	Hou et al. [15]
0.6	0.9813	0.6593	0.7120
0.7	0.9407	0.5573	0.6884
0.8	0.8553	0.4943	0.7061
0.9	0.6960	0.4490	0.6876
1.1	0.7413	0.6247	0.6902
1.2	0.8127	0.7027	0.6657
1.3	0.9093	0.7090	0.6603
1.4	0.9433	0.7617	0.6523
1.5	0.9649	0.7873	0.6346
1.6	0.9791	0.8373	0.6804
1.7	0.9902	0.8910	0.7725
1.8	0.9947	0.9447	0.8484
1.9	0.9976	0.9773	0.8863
2.0	0.9987	0.9780	0.8696

Figure 9. Comparison of the accuracy of p/q from 0.6 to 2.

The following observations can be made from Table 4 and Figure 9:

- (1) The detection accuracy is improved notably with the increase of the up-resampling factor. This is because the local texture difference between the original and resampled speech becomes more obvious as the compression degree of the spectrogram increases.
- (2) The detection performance degrades when the resampling factor is close to 1.0. The potential reason is that the local texture difference between the original and resampled speech is getting smaller and smaller, which makes the resampled speech hard to detect.
- (3) Among these three methods, the proposed method always outperforms the other two methods, which demonstrates that the proposed features are more discriminative for speech resampling detection, especially in small-scale resampling.

5.3. Experimental Results under Different Compression Scenarios

MP3 compression has become the most popular compression format in recent years. The compression principle is that low-frequency speech carries much information and compression removes some of the redundant information [24,25]. Therefore, the speech compressed by MP3 has no obvious distortion. The resampled speech is often compressed by MP3 in the transmission process. As a result, the detection of speech resampling under compression is meaningful in speech forensics. The higher the MP3 compression bit rate, the better the speech quality and vice versa. This paper used the standardized MPEG-I layer 3 codec to carry out the experiments [26] with the LAME tool. Three commonly used compression bit rates, that is, 32 kbps, 64 kbps, and 128 kbps, were investigated in this

paper to evaluate the robustness of the algorithms. The detection rates with the proposed method and the two investigated methods are shown in Table 5 and Figure 10.

As can be seen in Table 5 and Figure 10, the proposed detector also worked well against Mp3 compression. The performance of the proposed detector is still better than that of the other two detectors, which indicates that the proposed detector is more robust in MP3 compression scenarios. Furthermore, when the compression bit rate is fixed, the detection result is better with the increase of the up/down sampling scale. Moreover, it can be seen that the detection rate is slightly improved with the increase in the compression level. Since interpolation introduces some residual traces in the high-frequency component of the spectrogram, the details related to the high-frequency information are weakened after compression. Hence, the partial influence introduced by interpolation is offset in the high-frequency component [27], which slightly increases the detector's performance.

Figure 10. Influence of different compression rates on speech resampling detection. (a) The detection accuracy of the MP3 compression bit rate at 32 kbps; (b) the detection accuracy of the MP3 compression bit rate at 64 kbps; (c) the detection accuracy of the MP3 compression bit rate at 128 kbps.

Table 5. Detection results under different compression bit rates.

p/q	Proposed Method at 32 kbps	Wang et al. [16] at 32 kbps	Hou et al. [15] at 32 kbps	Proposed Method at 64 kbps	Wang et al. [16] at 64 kbps	Hou et al. [15] at 64 kbps	Proposed Method at 128 kbps	Wang et al. [16] at 128 kbps	Hou et al. [15] at 128 kbps
0.6	0.9980	0.6593	0.6984	0.9860	0.6330	0.7059	0.9860	0.6370	0.7179
0.7	0.9787	0.5447	0.6977	0.9407	0.5297	0.7219	0.9373	0.5273	0.7346
0.8	0.9627	0.4977	0.6968	0.8473	0.4983	0.7061	0.8440	0.4993	0.6963
0.9	0.9500	0.4387	0.7222	0.7040	0.4677	0.6702	0.6847	0.4587	0.6795
1.1	0.9540	0.6683	0.7303	0.7380	0.6367	0.6800	0.7140	0.6397	0.6809
1.2	0.9580	0.7187	0.7485	0.8273	0.6833	0.6752	0.8187	0.6840	0.6884
1.3	0.9633	0.7433	0.6549	0.9047	0.7097	0.6482	0.8980	0.7097	0.6473
1.4	0.9760	0.7733	0.6596	0.9480	0.7587	0.6596	0.9407	0.7513	0.6464
1.5	0.9813	0.8110	0.6510	0.9733	0.7943	0.6407	0.9727	0.7973	0.6299
1.6	0.9913	0.8660	0.6393	0.9880	0.8400	0.6739	0.9840	0.8403	0.6543
1.7	0.9960	0.9000	0.7939	0.9960	0.8913	0.7901	0.9920	0.8877	0.7803
1.8	0.9980	0.9547	0.8708	0.9993	0.8437	0.8377	0.9987	0.9430	0.8642
1.9	0.9987	0.9757	0.9148	0.9993	0.9737	0.8862	0.9980	0.9763	0.8796
2.0	0.9993	0.9747	0.8900	0.9987	0.9773	0.8899	0.9993	0.9770	0.8815

6. Conclusions

This paper presented a novel forensic method for speech resampling identification. Since the change in the spectrogram is regular with the speech resampling scale, we theoretically analyzed the corresponding relationship between time domain and frequency domain of the resampled speech. For the above analysis, an LBP operator was used to model the local statistical change in the spectrogram caused by speech resampling. Experimental results showed that the proposed method is universal in different interpolation methods and robust to MP3 compressions. Compared with some existing methods, the detection performance was improved notably under various sampling factors, especially when the sampling scale was small. In future work, we will focus on feature optimization to further improve the performance of the algorithm.

Author Contributions: Conceptualization, Y.Z. and S.D.; methodology, Y.Z.; software, Y.Z. and S.D.; validation, Y.Z.; resources, Y.Z.; writing—original draft preparation, Y.Z. and S.D.; writing—review and editing, Y.Z., W.S., L.Z., and D.L.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was mainly funded by the Natural Science Foundation of Shanghai (grant no. 17ZR1411900), the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (grant no. AGK2015006), the Funding Program for the Cultivation of Young University Teachers of Shanghai (grant no. ZZGCD15090), and the Research Start-Up Foundation of Shanghai University of Engineering Science (grant no. 2016-56).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Li, F.; Xin, M.; Li, J.; Yu, J. Improved detection for copy-move forgery with multi-scale sliding windows. In Proceedings of the 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Xiamen, China, 6–9 November 2017; pp. 231–236.
- Chen, C.L.; Ni, J.Q.; Shen, Z.Y.; Shi, Y.Q. Blind Forensics of Successive Geometric Transformations in Digital Images Using Spectral Method: Theory and Applications. *IEEE Trans. Image Process.* **2017**, *26*, 2811–2824. [[CrossRef](#)] [[PubMed](#)]
- Bappy, J.H.; Simons, C.; Nataraj, L.; Manjunath, B.S.; Roy-Chowdhury, A.K. Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries. *IEEE Trans. Image Process.* **2019**, *28*, 3286–3300. [[CrossRef](#)] [[PubMed](#)]
- Tang, W.; Li, B.; Tan, S.; Barni, M.; Huang, J. CNN-Based Adversarial Embedding for Image Steganography. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2074–2087. [[CrossRef](#)]

5. Ali, Z.; Imran, M.; Alsulaiman, M. An Automatic Digital Audio Authentication/Forensics System. *IEEE Access* **2017**, *5*, 2994–3007. [[CrossRef](#)]
6. Luo, D.; Korus, P.; Huang, J.W. Band energy difference for source attribution in audio forensics. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2179–2189. [[CrossRef](#)]
7. Luo, W.Q.; Li, H.D.; Yan, Q.; Yang, R. Improved audio steganalytic feature and its applications in audio forensics. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2018**, *14*, 1–14. [[CrossRef](#)]
8. Lin, X.; Kang, X. Exposing speech tampering via spectral phase analysis. *Digit. Signal Process.* **2017**, *60*, 63–74. [[CrossRef](#)]
9. Yan, Q.; Yang, R.; Huang, J. Robust Copy–Move Detection of Speech Recording Using Similarities of Pitch and Formant. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2331–2341. [[CrossRef](#)]
10. Yan, D.; Wang, R.; Zhou, J.; Jin, C.; Wang, Z. Compression history detection for MP3 audio. *Ksii Trans. Internet Inf. Syst.* **2018**, *12*, 662–675.
11. Yao, Q.M.; Chai, P.Q.; Xuan, G.R.; Yang, Z.Q.; Shi, Y.Q. Audio re-sampling detection in audio forensics based on EM algorithm. *J. Comput. Appl.* **2006**, *26*, 2598–2601.
12. Chen, Y.X.; Xi, W.U. A method of detecting re-sampling based on expectation maximization applied in audio blind forensics. *J. Circuits Syst.* **2012**, *17*, 118–123.
13. Shi, Q.; Ma, X.H. Detection of audio interpolation based on singular value decomposition. In Proceedings of the 2011 3rd International Conference on Awareness Science and Technology (iCAST), Dalian, China, 27–30 September 2011; pp. 287–290.
14. Qi, D.; Peng, X.J. Audio Tampering Detection Based on Band-Partitioning Spectral Smoothness. *Appl. Sci. Electron. Inf. Eng.* **2010**, *28*, 142–146.
15. Hou, L.M.; Wu, W.Q.; Zhang, X.P. Audio re-sampling detection in audio forensics based on second-order derivative. *J. Shanghai Univ.* **2014**, *20*, 304–312.
16. Wang, Z.F.; Yan, D.; Wang, R. Speech resampling detection based on inconsistency of band energy. *CMC-Comput. Mater. Contin.* **2018**, *56*, 247–259.
17. Wang, F.; Li, C.; Tian, L. An algorithm of detecting audio copy-move forgery based on DCT and SVD. In Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 1652–1657.
18. Mulimani, M.; Koolagudi, S.G. Segmentation and characterization of acoustic event spectrograms using singular value decomposition. *Expert Syst. Appl.* **2019**, *120*, 413–425. [[CrossRef](#)]
19. Mulimani, M.; Koolagudi, S.G. Extraction of MapReduce-based features from spectrograms for audio-based surveillance. *Digit. Signal Process.* **2019**, *87*, 1–9. [[CrossRef](#)]
20. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [[CrossRef](#)]
21. Liu, X.; Xue, F.; Teng, L. Surface Defect Detection Based on Gradient LBP. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 133–137.
22. Song, T.; Li, H.; Meng, F.; Wu, Q.; Cai, J. LETRIST: Locally Encoded Transform Feature Histogram for Rotation-Invariant Texture Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 1565–1579. [[CrossRef](#)]
23. Chang, C.; Lin, C. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2007**, *2*, 62–71. [[CrossRef](#)]
24. Wang, K.; Li, C.; Tian, L. Audio zero watermarking for MP3 based on low frequency energy. In Proceedings of the 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMT), Himeji, Japan, 1–7 September 2017; pp. 1–5.
25. Erfani, Y.; Pichevar, R.; Rouat, J. Audio Watermarking Using Spikegram and a Two-Dictionary Approach. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 840–852. [[CrossRef](#)]

26. Brandenburg, K.; Stoll, G. ISO/MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio. *J. Audio Eng. Soc.* **1994**, *42*, 780–792.
27. Zhang, Q.Y.; Zhang, T.; Qiao, S.B.; Wu, D.F. Spectrogram-based Efficient Perceptual Hashing Scheme for Speech Identification. *Int. J. Netw. Secur.* **2019**, *21*, 259–268.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).