Blind Microphone Analysis and Stable Tone Phase Analysis for Audio Tampering Detection

Luca Cuccovillo¹, Sebastian Mann¹, Patrick Aichroth¹, Marco Tagliasacchi² Christian Dittmar¹

¹Fraunhofer Institute for Digital Media Technology, Ilmenau, 98693, Germany
²Politecnico di Milano, Milano, 20133, Italy

Correspondence should be addressed to Sebastian Mann (sebastian.mann@idmt.fraunhofer.de)

ABSTRACT
In this paper, we present an audio tampering detection method, based on the combination of blind microphone analysis and phase analysis of stable tones, e.g., the electrical network frequency (ENF). The proposed algorithm uses phase analysis to detect segments that might have been tampered: These segments are further analyzed using a feature vector able to discriminate among different microphone types. Using this combined approach, it is possible to achieve a significantly lower false-positive rate and higher reliability as compared to standalone phase analysis.

1. INTRODUCTION
If recordings are tampered to convey a specific, distorted message, e.g. to be presented as forged evidence that a specific dialogue has taken place, it is often necessary to pick, cut and paste audio material from different sources, recorded using different devices and microphones (in the following, we assume that the microphone is the most relevant component for identifying recording devices). This circumstance can be exploited to improve on existing audio tampering detection approaches: Phase analysis of stable tones present in the audio file, such as Electrical Network Frequency (ENF) or sounds created by other devices or by the environment, can be used to indicate possible tampering. However, such approaches exhibit a low precision, as phase discontinuities can have other causes than tampering. By combining phase analysis approaches with microphone discrimination, which introduces another indication for tampering, overall detection rate can however be significantly improved. The algorithm proposed in this paper shows how microphone analysis can be adapted and combined with phase analysis, to achieve this goal.
Stable tone phase analysis is a common approach to audio tampering detection. Such analysis typically starts with Short Time Fourier Transform, which can be applied to all kinds of stable tones possibly present in the audio recording, and is followed by an analysis of the phase of the extracted stable signal. The underlying assumption is that editing operations upon audio material introduces phase discontinuities, which are difficult to avoid during tampering, and which can be detected. Problem is, such discontinuities are not only introduced by tampering, but also due to natural variations, technical faults or computing errors, causing high false-positive rate for detection. Microphone classification, on the other hand, was first investigated in 2007 (see [1]), providing a proof of concept of microphone discrimination both for clustering procedures, and for classification algorithms. Current state-of-the-art approaches to microphone classification are based on a statistical characterization of the devices, achieved by appending the means of a Gaussian mixture that has been trained on the known device recordings. The results of such approaches are very good, with an average accuracy beyond 90%, but the applicability of these approaches for tampering detection is limited due to the closed-set assumption: In many cases, the microphones involved in the recording are unknown, and thus no respective training of the algorithms is possible. This paper is structured as followed: Section 2 describes the stable tone phase analysis, while Section 3 outlines the chosen microphone characterization. The proposed tampering detection algorithm is presented in Section 4 and evaluated in Section 5. Section 6 concludes with an outlook on possible improvements to the algorithm.

2. STABLE TONE PHASE ANALYSIS

The stable tone phase analysis consists of two parts, the extraction of the stable tone from the signal and the analysis of the phase. The first consists of obtaining the phase from the stable tone, while the latter addresses the tampering detection.

2.1. Stable tone extraction

The prerequisite for the phase analysis of a stable tone is the extraction from the signal. In the literature there are mainly two different ways to extract a stable tone (as shown in [2], [3], [4], [5] and [6]). The literature focuses on the extraction the ENF from a given signal, but all of the methods are applicable to all kind of stable tones, once the target frequency has been chosen. The first methods are time-based, and measure the frequency by determining the period of the stable tone oscillation (e.g. via zero crossing methods) and by applying different filters and interpolation methods. The second ones are frequency-based methods, which are mostly using the short-time Fourier transform (STFT), that operates on overlapping or adjacent frames of the audio signal. Sensible parameters are the length of the transform, the window function and the step size.

Each technique has an application context where it can achieve the best accuracy. Time-based methods are useful to extract the ENF directly from the power line, but not for real-world content. Frequency-based methods are suitable for real-life speech or music audio content, and can safely address all kind of stable tones. In the context of this paper we are using frequency/STFT-based methods for extracting the stable tones, since they are the most suitable when handling real-life recordings. The extraction also requires some pre-processing, e.g. bandpass filtering and down-sampling.

2.2. Phase Analysis

After extracting the stable tone from the signal, it is necessary to obtain the phase and then analyze it in order to discover some discontinuities. Each of the detected discontinuities within the phase can indicate a tampering point, since it is almost impossible to merge two audio parts so that the same stable tones are blending into each other in a mathematically continuous way. Unfortunately it is possible that discontinuities can also occur due to other reasons than tampering. They are caused by technical problems within the power grid, e.g. sudden peaks in energy consumption, or by extraction errors. To obtain the phase of the extracted stable tone and thus find tampered regions, a Discrete Fourier transform (DFT) of the given and windowed signal \( x(n) \) is used, where

\[
X(k) = DFT(x(n)),
\]

as proposed in [7]. The phase is then simply obtained by getting the argument of \( X(k) \):

\[
\Phi_{\text{ENF}} = \angle(X(k_{\text{peak}})),
\]

where

\[
k_{\text{peak}} := \arg \max_{k \in K} |X(k)|,
\]

and \( K \) is the number of frequency bins. The obtained phase of a tampered file, where a segment of another recording was spliced into the original one, is shown in Figure 1, where the purple curve denotes the theoretically phase, blue is the extracted phase and the red (max.) and green (min.) circle denotes the extreme values of the
phase.
The detection method is described below:

1. Separating the phase into intervals. One interval is
   the phase between a minimum value (green) and the
   following maximum value (red) of the phase.

2. Computing the mean squared error (MSE) of the di-
   stance between the theoretically and the extracted
   phase for each interval.

3. Selecting the intervals where the MSE exceeds a gi-
   ven threshold, and marking them as a possible tam-
   pering points.

The MSE results for the phase in Figure 1 are shown in
Figure 2, where the red line is the given threshold and
the two intervals exceeding the threshold will be marked
as possible tampering points. By providing such suspi-
cious intervals, it is possible to localize the tampering: If
a segment of another audio file was spliced into the ori-
ginal one, we can detect the start and end frame of the
spliced segment. As can be seen in Figure 2, however,
when lowering the threshold, discontinuities do not only
appear in tampered, but also in non-tampered segments,
due to the named reasons. Whenever this happens, the
stable tone phase analysis incorrectly classifies the inter-
val as being tampered, leading to a high number of false
positive detections. In order to tackle this problem, a fur-
ther microphone analysis can be performed. The feature
vector involved in the analysis is described in the follo-
wing section.

3. MICROPHONE ANALYSIS

The microphone analysis algorithm is based on feature
vector derived from a recent work by Gaubitch et al. [8],
where the authors were trying to model the channel
representing the environment in which the acoustic
signal propagates.

In our method, however, the channel models the fre-
quency response of the microphone, and the resulting
feature vector is designed to work on real-life recordings
with environmental noise and music present, whereas [8]
refers to a noiseless speech-only content.

3.1. Training of a Gaussian Mixture Model for
Clean Speech

In order to successfully obtain a meaningful channel
estimate, we need to train a Gaussian Mixture Model
(GMM). This GMM provides a general model of the
clean speech signal\footnote{Ideal speech signal which has not undergone any spectral modification}, starting from noiseless recordings
of sentences produced by female and male speakers.

The GMM represents $M$ classes of average speech
spectra. In order to compute it, we need a suitable set
of features to represent a frame of the speech spectrum.
Ideally, the features should not be affected by the
channel. [8] states that a suitable candidate are the
RASTA filtered Mel-Frequency Cepstral Coefficients
(RASTA-MFCC), which were developed as robust
features for speech recognition: RASTA processing,
indeed, performs bandpass filtering in the cepstral
domain in order to reduce channel effects [9].

The training of the GMM completely follows the algo-

rithm detailed in [8]. The training allows us to obtain the
mean $\mu_i$, diagonal covariance $\Sigma_i$ and weight $\pi_i$ of each
mixture, as well as a matrix $\hat{Z}_S \in \mathbb{R}^{M \times N_m}$ that denotes
the average log spectrum of the speech. The $i$-th row of
$\hat{Z}_S$ represents the average log-spectrum corresponding to
the $i$-th mixture of the GMM, and the $j$-th column
represents a single frequency bin. The total number of
frequency bins is denoted by $N_m$.

3.2. Blind Channel Estimation

In order to perform the blind channel estimation the test
file $x(n)$ is split into frames $x_i(n)$. Afterwards, differently from [8], a subset $L_x$ from the original set of $L_x$ frames of the test file is selected, by following
\[ \hat{l} \in L_x \iff \max_{k \in N_m}(|X_i(k)|) \leq \tau \cdot \left( \max_{\ell \in L_x, k \in N_m}(|X_\ell(k)|) \right) \]
where $X_i$ denotes the short-term Fourier Transform of the frame $x_i(n)$, $N_m$ denotes the number of frequency bins, and $\tau$ is a user-defined threshold. For each frame $\hat{l} \in L_x$ we compute a feature vector $c_{X,\hat{l}}$ of $N_c$ RASTA-MFCC coefficients. The parameters $(\pi_i, \mu_i, \Sigma_i)$ of the GMM are used in order to compute a relative probability matrix $P_X \in \mathbb{R}^{M \times L_x}$ as follows:
\[ P_X = \left( p\left(z_i = 1|c_{X,\hat{l}}\right) \right), \]
\[ i = 1, \ldots, M, \hat{l} \in L_x \]
where
\[ p\left(z_i = 1|c_{X,\hat{l}}\right) = \frac{\pi_i \cdot N\left(c_{X,\hat{l}}|\mu_i, \Sigma_i\right)}{\sum_{m=1}^{M} \pi_m \cdot N\left(c_{X,\hat{l}}|\mu_m, \Sigma_m\right)} \]
z$_i \in \{0,1\}$ is the selection variable for the $i$-th component of the GMM. By using the relative mixture probability matrix $P_X$ as a selection matrix on the average log spectrum of the speech $\hat{Z}_S$, we are able to estimate the non-filtered ideal speech source of the test recordings:
\[ \hat{Z}_X = P_X^t \cdot \hat{Z}_S \]
where the superscript $t$ denotes the matrix transpose. The estimate $\hat{h}$ of the magnitude response of the original channel $h$ can be computed as the difference between the average log spectrum of the test file and the estimate of the original clean speech:
\[ \hat{h} = \left( Z_X - \hat{Z}_X \right)^t \cdot \frac{1}{L_X} \] (1)
where $1$ is a $L_X \times 1$ vector with all elements equal to one and the superscript $t$ denotes the matrix transpose. $\hat{Z}_X$ can be computed as follows [8]: The log spectrum of each selected frame $x_\ell(n)$ is normalized by subtracting its mean
\[ Z_{X,\ell} = \log(|X_\ell|) - \frac{1}{N_m} \sum_{k=1}^{N_m} \log(|X_\ell(k)|), \forall \ell \in L_x \] (2)
and all the normalized log power spectra are used to build the matrix $Z_X$, where $Z_X \in \mathbb{R}^{L_X \times N_m}$.

### 3.3. Feature Vector Computation

For a test audio file $x(n)$ we compute three multidimensional features, denoted in the following as $f_1$, $f_2$ and $f_3$. These three features, despite being similar to each other, have different meanings: In $f_1$ we keep all the information obtained by the channel estimation algorithm; $f_2$ is meant to be a descriptor of the correlation between the channel estimate and the original spectrum of the audio file; $f_3$ defines the properties of the approximated spectrum of the audio file.

Let $Z_X$ be the normalized power of $x(n)$ defined in equation (2), $\hat{h}$ the channel estimate of $x(n)$ defined in equation (1) and
\[ \hat{p} = \frac{(Z_X)^t \cdot 1}{L_X} \]
an approximation of the average power, where $1$ is a $L_X \times 1$ vector with all elements equal to one and the superscript $t$ denotes the matrix transpose. Let $(v)$ denote the average value of a generic vector $v$. We can define the first and second derivative of this generic vector $v$ of length $D$ by means of the inter-component difference:
\[ v'(d) = v(d) - v(d-1) \quad \forall d \in [2, D], \]
\[ v''(d) = v'(d) - v'(d-1) \quad \forall d \in [3, D]. \]

The definition of the feature $f_1$ is the following:
\[ f_1 = \left[ h_1, h'_1, h''_1 \right], \]
with $h_1 = \gamma_1 \cdot \left( \hat{h} + \frac{\hat{p}}{\hat{p}} \right)$
where $\gamma_1$ is a variable gain driven by the power of $\hat{h}$. As we can see, $f_1$ keeps all the information about the channel estimate. $f_2$ is a descriptor of the correlation between the channel estimate and the original spectrum of the audio file, and can be defined as follows:
\[ f_2 = \left[ h_2, h'_2, h''_2 \right], \]
with $h_2 = \gamma_2 \cdot \left( \frac{\hat{h}}{\hat{p}} \right)$
where ./ denotes the entry-wise division$^2$ between two elements from the same space, and $\gamma_2$ is a variable gain
\[ \text{AE}^2C := \frac{A_i \cdot B_i}{C_i}, \text{ with } C_{i,j} = A_{i,j} / B_{i,j}, \forall (i,j) \in I \times J, A, B, C \in \mathbb{R}^{I \times J} \]
driven by the power of $\hat{h}$ and $\hat{p}$. $f_1$ and $f_2$ are processed in order to fully enhance their descriptive power. The processing of $f_3$, on the other hand, is also meant to eliminate as much as possible redundant information between $f_1$ and $f_3$, and to ensure that the influence due to the content of the recording on $\hat{p}$ is not strong enough to affect the outcome of the classification. $f_3$ is computed as follows:

$$f_3 = \left[ \text{norm}_{[0,1]}(h_3), \text{norm}_{[0,1]}(\gamma_3 \cdot h_3), \text{norm}_{[0,1]}(\gamma_3 \cdot |h_3|) \right],$$

with $h_3 = \gamma_3 \cdot (\hat{p} + \overline{h})$,

where $\gamma_3$, $\gamma_3$, and $\gamma_3$ are variable gains driven by the power of $\hat{p}$, and $\text{norm}([0,1])$ is the function defined as follows:

$$\text{norm}([0,1])(\cdot) = \frac{(\cdot) - \min(\cdot)}{\max(\cdot) - \min(\cdot)}.$$

The final feature vector $f$ is computed by collecting the three features:

$$f = [f_1, f_2, f_3]. \tag{3}$$

A visual comparison of $f_1$, $f_2$, and $f_3$ is shown in Figure 3, where the x-axis represents the dimension index. The feature vector embeds an high-level representation of the microphone involved during the recording. We will see how to exploit this representation in Section 4.

4. TAMPERING DETECTION ALGORITHM

The combination of stable tone analysis and microphone analysis is achieved by first searching for suspicious intervals within the audio recordings, using the phase analysis to detect discontinuities which are used to segment the recording. Afterwards, those intervals are analyzed with a microphone discrimination algorithm in order to check whether different microphones were used during the recording or not. If different microphones are detected, those intervals are marked as tampered and the corresponding discontinuities are the tampering borders, which locate the tampering. Discontinuities within section of the same microphone are rejected as false-positives. The process flow of the tampering detection algorithm is shown in figure 4 and described below in a more formal way.

1. In the first step the stable tone is extracted from the original signal.

2. After that, the phase analysis is performed, as described in Section 2. The phase of embedded stable stones (such as ENF) is obtained by computing the discrete Fourier transform of the windowed signal. Discontinuities are then localized by analyzing the phase difference between each two consecutive frames. If a given threshold is exceeded, the frame is marked as a discontinuity, serving as a tampering segment border.

3. Those borders are used to segment the recording. Thanks to the previous phase analysis, this segmentation can be already considered to be tampering-aware.

4. The microphone analysis detailed in Section 3 is performed once per each segment. Each feature vector can be post-processed, e.g. by the means of a

---

3 the features are computed only for the frequency bins from 1 to $N_{\text{fft}}/2$, in order to avoid redundancies
feature selection, and the microphone is finally characterized by a single point in a multidimensional space.

5. If the correlation coefficient ($r_{XY}$) between the feature vectors of two adjacent segments is less than a given threshold, the tampering suspicion and location are confirmed.

6. The final tampering locations are determined by the intersection of the tampering locations determined via phase analysis and the ones determined via microphone analysis and discrimination.

The correlation coefficient is thus used as a similarity measure, and is defined as follows:

$$r_{XY} = \frac{1}{n-1} \sum_{i=1}^{n} \left( \frac{X_i - \bar{X}}{s_X} \right) \left( \frac{Y_i - \bar{Y}}{s_Y} \right),$$

where $s_X$ and $s_Y$ denotes the standard deviation of the vectors $X$ and $Y$, both of length $n$. $\bar{X}$ and $\bar{Y}$ are the mean value of the vectors.

This combined approach is meant to detect inconsistencies due to different microphones involved, thus avoiding the high false-positive-rate exhibited by phase analysis alone. Moreover, the aforementioned limitations due to the closed-set assumption of microphone classification algorithms are removed, since the microphone analysis does not require any previous knowledge of the device involved. A description of the test sets involved and of the related results can be found in Section 5.

### 5. TAMPERING DETECTION RESULTS

#### 5.1. Test Set Generation

The tampered test content was created by splicing a short interval of a recording from one device into one recording originally acquired by another device. In order for the tampering not to be completely obvious, e.g. due to a change in the environmental noise, the original recordings were acquired by arranging the four mobile phones in a microphone-array configuration. The microphones involved in the test content generation are shown in Table 1. Starting from six original recordings per device, the procedure mentioned above generated 216 tampered recordings. Depending on the family of the device used in order to record the spliced interval, we can identify three subsets.

<table>
<thead>
<tr>
<th>Family</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Samsung Galaxy S III, built-in microphone</td>
</tr>
<tr>
<td>2</td>
<td>iPhone 3GS, built-in microphone</td>
</tr>
</tbody>
</table>

#### 5.2. Evaluation

Our main focus in the evaluation procedure was to correctly assess the reliability of a tampering detection per-
Table 3: Tampering Detection Results

(a) Related Devices

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Outcome (#)</th>
<th>Statistics (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP  FP  TN  FN</td>
<td>Precision  Recall  FP rate  Accuracy</td>
</tr>
<tr>
<td>Phase</td>
<td>140  86  34  4</td>
<td>61.95  97.22  71.67  65.91</td>
</tr>
<tr>
<td>Combined</td>
<td>100  6  114  44</td>
<td>94.34  69.44  5.00  81.06</td>
</tr>
</tbody>
</table>

(b) Unrelated Devices

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Outcome (#)</th>
<th>Statistics (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP  FP  TN  FN</td>
<td>Precision  Recall  FP rate  Accuracy</td>
</tr>
<tr>
<td>Phase</td>
<td>277  86  34  11</td>
<td>76.31  96.18  71.67  76.23</td>
</tr>
<tr>
<td>Combined</td>
<td>172  6  114  116</td>
<td>96.62  59.72  5.00  70.10</td>
</tr>
</tbody>
</table>

(c) General Use Case

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Outcome (#)</th>
<th>Statistics (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP  FP  TN  FN</td>
<td>Precision  Recall  FP rate  Accuracy</td>
</tr>
<tr>
<td>Phase</td>
<td>417  86  34  15</td>
<td>82.90  96.53  71.67  81.70</td>
</tr>
<tr>
<td>Combined</td>
<td>272  6  114  160</td>
<td>97.84  62.96  5.00  69.93</td>
</tr>
</tbody>
</table>

formed by combining the stable tone phase analysis and the microphone discrimination, in comparison with a tampering detection performed with the phase analysis by itself.

The phase discontinuity procedure was performed on a stable tone with a mean frequency values of 47.6 Hz, with an high magnitude throughout the duration of the whole test content. After collecting the data, from each detected interval the feature vector was computed, as described in section 4, and compared with its neighbors.

It was finally possible to compute the Receiver Operating Characteristic (ROC) curve of the tampering detection procedure. In Figure 5 we reported three ROC curves, one per each test subset: related devices (red line, symbol ‘×’); unrelated devices (green line, symbol ‘+’); general use case (blue line, symbol ‘○’). As common, the X-axis represent the False Positive rate and the Y-axis represents the True Positive rate.

Differently from the common ROC curves, the True Positive rate and the False Positive rate are not able to reach the value of 1, due to an upper bound created by the detection capability of the underlying phase analysis.

From the ROC curves it is also possible to notice how, by lowering the detection threshold of the microphone discrimination algorithm, the tampering detection reliability increase: The starting False Positive rate of ≈ 70%, which can be considered to be completely unacceptable, can safely drop to 5-10% while keeping a True Positive rate of 60-70%.

The detailed results for each test set at the operating point that guarantees the highest accuracy with a False Positive Rate of 5% are reported in Table 3a, 3b and 3c, where True Positive indicates that a tampered recording was correctly classified as tampered and True Negative indicates that an authentic recording was correctly classified as non-tampered.

Considering that the involved devices share the same quality and purpose (see Table 1 and 2) and that also non-trivial discrimination tasks between highly related microphone were present, due to the ad-hoc tampered test content generation, we believe that the proposed method can be considered to be sufficiently reliable. The initial goal of overcoming the high False Positive rate inherent the phase analysis as a stand-alone procedure can be safely considered to have been met. Moreover, the precision achieved for all the three test sets is higher than 90%,
even for the most challenging test set which contains highly related devices. The recall, on the other hand, is still far from being satisfying: An outlook on possible improvements to the algorithm, which also tackle this problem, can be found in section 6

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new combined approach for tampering detection, by combining stable tone phase analysis and microphone analysis. This method overcomes both the high false positive rate peculiar of phase-discontinuity based tampering detection method, and the limited applicability of State-of-the-Art microphone classification methods. Moreover, its applicability is not limited to recordings where the ENF is present.

The results prove that, by reducing the overall recall, it is possible to reduce the false positive rate from an initial value of 71.67% down to 5%. The precision across the three test sets has been greatly improved, and is always higher than 90%. Hence, we believe that the overall reliability of the detection increased consequently.

The low recall of the audio tampering detection algorithm proposed has still room for improvements, e.g., a better combination of the stable tone analysis and microphone analysis should consider the possibility that phase discontinuities exceeding a defined magnitude should not be dropped, and a different feature vector or a more refined blind channel estimation of the microphone frequency response would result in an higher discrimination capability.

Besides the necessary improvements on the algorithm by itself, and interesting outlook could be the possibility to combine this method with completely different kinds of audio tampering detection techniques.

ACKNOWLEDGMENTS

This research has been partially funded by the EU project REWIND. The project acknowledges the financial support of the Future and Emerging Technologies (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 268478.

7. REFERENCES


