INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Blind Channel Response Estimation for Replay Attack Detection

Anderson R. Avila[1,2], Jahangir Alam[2], Douglas O'Shaughnessy[1], Tiago H. Falk[1]

[1]INRS-EMT, University of Quebec, Canada
[2]Computer Research Institut of Montreal (CRIM), Canada

anderson.avila@emt.inrs.ca, jahangir.alam@crim.ca, dougo@emt.inrs.ca, falk@emt.inrs.ca

Abstract

Recently, automatic speaker verification (ASV) systems have been acknowledged to be vulnerable to replay attacks. Multiple efforts have been taken by the research community to improve ASV robustness. In this paper, we propose a replay attack countermeasure based on the blind estimation of the magnitude of channel responses. For that, the log-spectrum average of the clean speech signal is predicted from a Gaussian mixture model (GMM) of RASTA filtered mel-frequency cesptral coefficients (MFCCs) trained on clean speech. The magnitude response of the channel is obtained by subtracting the log-spectrum of the observed signal from the predicted log-spectrum average of the clean signal. Two datasets are used in our experiments: (1) the TIMIT dataset, which is used to train the log-spectrum average of the clean signal; and (2) a dataset containing replay attacks used during the second Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2017). Performance is compared to two benchmarks. The discrete Fourier transform power spectral (DFTspec) and the constant Q cepstral coefficients (CQCCs). Results show the proposed method outperfoming the two benchmarks in most scenarios with equal error rate (EER) as low as 6.87% when testing on the development set and as low as 11.28% on the evaluation set.

Index Terms: Automatic speaker recognition, spoofing attacks, replay attack, channel estimation

1. Introduction

In recent years, new advances in channel compensation have taken automatic speaker verification to the next level. Its deployment as a low-cost voice authentication solution (e.g., unlocking a smartphone with a voice command) has already become a reality, being currently used, for example, by a number of financial institutions [1]. This is driven mainly by the increased use of mobile devices as well as by the convenience and non-intrusiveness offered by such technologies. In fact, recent reports predict a continued growth of the mobile biometrics sector due to increased consumer demand for safety, especially while using mobile devices for banking transactions and e-commerce [1].

Despite all these advances, the vulnerability of ASV in the face of spoofing attacks (i.e., impersonation, replay attacks, speech synthesis and voice conversion) has become an increasing concern for the research community. Therefore, many initiatives to develop spoof countermeasures have been made lately [2][3]. Many of the efforts in this direction have been focused on developing anti-spoofing techniques to protect ASV systems against speech synthesis (SS) and voice conversion (VC) [2]. In this study, we are particularly interested in countermeasures to replay attacks, which consists of an attempt to fool an ASV system by playing-back a pre-recorded speech sample. In such circumstances, detecting the replay attack beforehand is crucial to maintain ASV reliability.

Hence, the problem has been recently addressed by the research community. The second and third Automatic Speaker Verification Spoofing and Countermeasures Challenges (ASVspoof 2017 and 2019) [3][4], for instance, provided common databases, protocols and metrics to evaluate countermeasure solutions and focused specifically on replay attacks. Although considered the easiest form of spoofing (e.g., no special expertise nor equipment is required [5]), to date only a few studies have addressed replay attacks when compared to other forms of spoofing. For example, in [6], the authors present a playback attack detector (PAD) based on a Gaussian mixture model (GMM) supervector (GSV) with a binary classifier based on a support vector machine (SVM). The authors in [7] rely on spectral bitmaps or spectral peaks, which are time-frequency points higher than a pre-defined threshold. The similarity score is attained by computing an element-wise product between the spectral bitmap of the verification sample and the stored spectral bitmaps. More recently, the performance of several features and classifiers is described by [5]. The authors report results from six magnitude spectrum and three phase spectrum based features on the ASVspoof 2017 replay attack detection challenge, with experiments revealing the superiority of the magnitude spectrum features over phase based features for all four classifiers tested. Despite all the advancements in this field, investigating new countermeasures solutions is still relevant.

In this work, we propose the use of blind channel spectrum estimation to detect replay attacks. Considering that in a replay attack the utterance will be acoustically affected by factors such as the recording, the room environment and the playback devices, it is expected that such affects will be encountered in the spectrum. We propose to estimate such variations in the spectrum by computing the magnitude response of the channel. This is achieved by first training a clean speech model based on a GMM. The model is trained using a RASTA filtered mel-frequency cepstral coefficients (MFCCs) extracted from a number of clean speech, allowing us to attain the log-spectrum average. By computing the log-spectrum average of clean signals and then subtracting it from the log-spectrum of the observed signal. Principal component analysis (PCA) is also used to reduce feature dimension and to boost performance. As a classifier, a simple GMM is adopted to distinguish between bonafide and spoof utterances. Results show the proposed method outperforming the benchmarks on both the development and evaluation set. Therefore, this work is an important contribution towards new spoofing countermeasure approaches as, to the best of our knowledge, no study on estimating channel response magnitude for replay attack detection has already been made.

The rest of this paper is organized as follows. Section II provides a description of the proposed method. In Section III, we present our experiment setup and Section IV discusses our experimental results. Section V concludes the paper.

http://dx.doi.org/10.21437/Interspeech.2019-2956

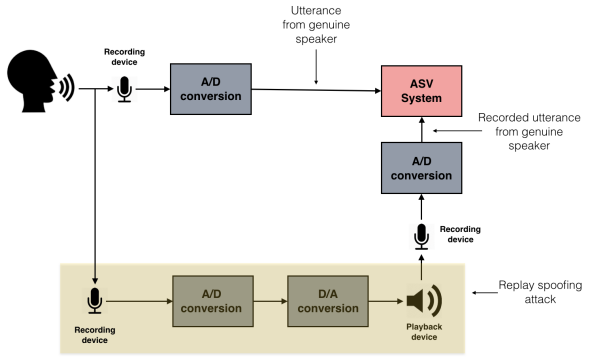

Figure 1: *Replay attack scenario.*

2. Proposed method

In this section, the proposed method is described, along with the steps to attain the average spectra of clean speech, followed by the explanation on how the channel response magnitude is estimated.

2.1. General Principles

Figure 1 illustrates a replay spoofing attack scenario. In such circumstance, an utterance recorded from the genuine speaker is presented to the ASV system. This utterance is likely to be acoustically affected by multiple replay configurations (RC), i.e., different combination of room environment, recording and playback devices. Hereafter, we refer to all the possible effects encompassing the recorded utterance captured by the ASV microphone as the channel effect, which can be expressed as:

$$x(n) = s(n) * h(n) + v(n), \tag{1}$$

where $s(n)$ represents the speech signal from a bonafide access attempt, $h(n)$ denotes the channel impulse response and $v(n)$ refers to the additive noise. By applying the short-time Fourier transform (STFT), we can write Eq. (1) as:

$$X(k,l) = S(k,l)H(k) + V(k,l), \tag{2}$$

with each frequency bin represented by k and the time frame by l. With prior knowledge of the log-magnitude spectrum of the clean speech signal, $S(k,l)$, the log-magnitude spectrum of the channel can be estimated as [8]:

$$\hat{\underline{H}}(k,l) = \frac{1}{L}\sum_{l=1}^{L}(\underline{X}(k,l) - \underline{S}(k,l)), \tag{3}$$

where $\underline{A}(k,l) = \log(|A(k,l)|)$, $\hat{A}(k,l)$ represents an estimate of $A(k,l)$, which denotes the STFT of the lth frame, with k being the frequency bin index [8]. Note that in a noiseless environment $V(k,l) \equiv 0$.

In the remainder of this section we discuss the steps to attain an estimate of the true log-magnitude spectrum based on training a clean speech model, such that $\hat{\underline{S}}(k,l) \approx \underline{S}(k,l)$. Following, we describe how we use the parameters of such a model to estimate the unknown channel response $\underline{\hat{H}}(k)$.

2.2. Log-Magnitude Spectrum from a Clean Speech Model

Figure 2 depicts the steps to attain the channel response spectrum magnitude. In the lower half, clean speech, $s(n)$, is used to train the clean speech model, based on a Gaussian mixture model (GMM). To achieve that, the speech signals are first segmented into frames of 32-ms length, with 16-ms hop-size. Prior to computing the STFT, the speech signals are pre-emphasized by a filter of coefficient 0.97, which is meant to balance low and high frequency magnitudes and after multiplied by a hanning window. The attained log-spectrum, $\underline{S}(k,l)$, is then normalized by subtracting the log-spectrum mean as follows:

$$\tilde{\underline{S}}(k,l) = \underline{S}(k,l) - \frac{1}{K}\sum_{k=1}^{K}\underline{S}(k,l), \tag{4}$$

where the number of STFT points is defined by K. Note that MFCCs are also computed. A total of 12 coefficients plus the log energy are attained, leading to a 13-dimensional vector for each frame. Finally, a RASTA filter is also applied to mitigate channel effects [8]. Then, once speech parametrization is performed, the MFCC-RASTA coefficients, $c_S(l)$, are used to train a Gaussian mixture model. The GMM adopted here contains 1024 Gaussians and mixture probabilities given by:

$$p_{l,m}(c_s(l)) = \frac{\pi_m \mathcal{N}(c_s(l)|\mu_m, \sum_m)}{\sum_{j=1}^{M}\pi_j \mathcal{N}(c_s(l)|\mu_j, \sum_j)}, \tag{5}$$

where $\lambda = \{\mu_m, \sum_m, \pi_m\}$ are the parameters of a multivariate Gaussian distribution denoted by $\mathcal{N}(c_s(l)|\mu_m, \sum_m)$. As our goal is to attain an average of the short-term log-spectra, $p_{l,m}(c_s(l))$ and $\overline{\underline{S}}(k,l)$ are combined over all available frames of the training data. This leads to M average clean speech log-spectra:

$$\overline{S}_m(k) = \frac{\sum_{l=1}^{L} p_{l,m}(c_s(l))\tilde{\underline{S}}(k,l)}{\sum_{l=1}^{L} p_{l,m}(c_s(l))}, \forall k, m = 1, \dots M. \tag{6}$$

where M is 1024. Note that each mixture is associated with a clean speech spectrum, attained from the weighted average of multiple clean speech spectra assigned to a particular mixture.

2.3. Channel Response Estimation

The upper part of Figure 2 provides a description of how to estimate the unknown channel response magnitude. Similarly to section 2.2, the speech signal, $x(n)$, is segmented into overlapping frames and the same pre-processing steps are taken prior to extracting the STFT, $\overline{\underline{X}}(k,l)$, and the MFCC-RASTA coefficients, $c_x(l)$. For a given speech signal, the clean log-spectrum is then obtained using the feature vectors (i.e., $c_x(l)$) and the GMM parameters ($\mu_m, \sum_m$ and π_m) computed during the training phase. Then, the probability that a feature vector $c_x(l)$ belongs to the m-th mixture can be computed as in (5), which leads to a probability, $0 < p_{l,m} < 1$, for each mixture $m = 1, , M$. This probability can be used to estimate the clean speech spectra of the l-th frame using the weighted average of the average clean-speech spectra, $\hat{S}_m(k,l)$.

$$\hat{\underline{S}}_m(k,l) = \sum_{m=1}^{M} p_{l,m}(c_x(l))\overline{\underline{S}}(k,l), \forall k. \tag{7}$$

Considering $\hat{\underline{S}}(k,l) \approx S(k,l)$, the channel response estimation, $\hat{\underline{H}}(k,l)$, can be computed according to (3). Models of $\hat{\underline{H}}(k,l)$ are then computed for genuine and for spoof speech files and used for replay detection, as detailed in Section 3.4.

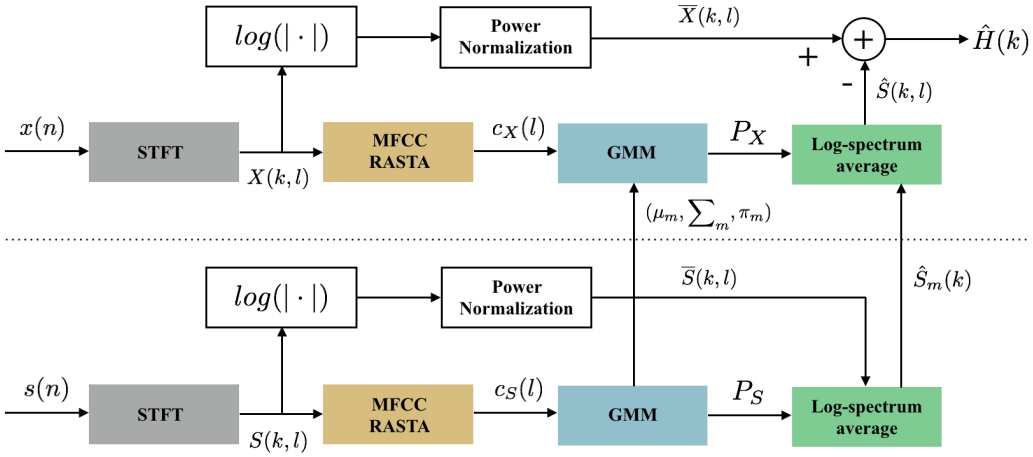

Figure 2: *Diagram for estimating the channel response spectrum magnitude*

3. Experimental Setup

In this section, we present the datasets used throughout the experiments, the adopted benchmarks, the classifier used for detecting replay attacks as well as the figure of merit.

3.1. Database Description

In our experiments, we used the well-known TIMIT and the ASVspoof 2017 datasets [9][3]. The TIMIT database has 630 speakers of eight different American English dialects. Initially developed for speech recognition applications, the database contains ten utterances recorded from each speaker. The data are divided into training and test sets. In this work, only the training set was considered. The TIMIT database was used to train the clean speech model used to estimate the average clean speech log-spectrum, as explained in Section 2.2. The second database used in our experiments is the text-dependent RedDots corpus [10] and its replayed version [3], which were adopted recently during the ASVspoof Challenge 2017 [3]. The dataset contains genuine and spoof replay recordings. In order to generate the replayed version of the RedDots corpus, genuine recordings were replayed in a variety of recording environments and playback devices. The data are partitioned into development, training and evaluation sets. The latter are used to evaluate the proposed and benchmark solutions when the models are trained using only the training set as well as both training and development sets.

3.2. Benchmark features

Two benchmarks features were considered in this work. The constant Q cepstral coefficients (CQCCs) were adopted as our primary benchmark feature. The reason is that these features have recently been introduced for spoofing attack detection in [11], and recurrently used as baseline in the most recent challenges involving spoof attack detection [3][4]. Derived from the perceptually-inspired constant Q transform (CQT), which has been widely used for music processing, the feature introduces the use of cepstral analysis on CQT. The CQCCs are attained by applying an inverse transformation to the discrete Fourier transform (DFT). Prior to extraction, the frequency bins of the CQT must be converted from geometric to linear space as an attempt to emulate the human auditory system [11]. This can be seen as a resampling operation as described in [12]. The CQCCs can be then extracted according to the equation below:

$$CQCC(p) = \sum_{l=1}^{L} \log \left| X^{CQ}(l)) \right|^2 \cos \left[\frac{p(l-\frac{1}{2})\pi}{L}. \right] \quad (8)$$

The Discrete Fourier transform (DFT) was the second benchmark feature adopted in order to compare the performance of our proposed method. These features have been successfully applied by the authors to the problem of spoofing detection in a recent work [13]. The processing steps to obtain the DFTspec features are straightfoward. For instance, given the speech signal $s(n)$, we consider the Fourier transform as:

$$Y(m,k) = \mathcal{F}\left(nx(m,n),\right) \quad (9)$$

where m is the frame index, n is the sample index and k is the frequency bin index. The power spectrum is $S(m,k) = |Y(m,k)|^2$. According to [13], improved results can be obtained if principal component analysis (PCA) is applied to the log power spectrum.

3.3. Dimensionality Reduction

Dimensionality reduction plays an important role in pattern recognition. Throughout our experiments, we used principal component analysis (PCA) to reduce dimensionality and boost performance. As in a high dimensional data many variables are interrelated, the method aims at finding a subspace of lower dimension where most of the variation and uncorrelated variables are ordered and kept in the few first dimensions. Therefore, PCA projection maximizes the variance of the projected points [14]. In our experiments, the principal components (PC) are learned from the training data and then the projection matrix is applied on the development and evaluation set. After that, we tested different feature dimensions and we empirically adopted 100-dimensional feature vector for the DFTspec and the proposed features and 40-dimensional feature vector for the CQCC. The former two had their dimension reduced from 256 and the latter from 90.

3.4. GMM Classifier

For the replay attack detection task, the score defined by the difference of log-likelihoods predicted by two Gaussian mixture models was used. We adopted 512 components for each of

the two models. The two mixture models are trained using the *Expectation-Maximization* algorithm independently on genuine and attack utterances. The described score for a given utterance represented by a feature vector $\mathbf{y}$ is defined in Eq. (10):

$$Score(\mathbf{y}) = \log \frac{P(\mathbf{y}|\lambda_g)}{P(\mathbf{y}|\lambda_s)}, \quad (10)$$

where λ_g and λ_s are the GMM models for genuine and spoof attack, respectively, and $P_g(\mathbf{x})$ and $P_s(\mathbf{x})$ are the likelihoods of $\mathbf{x}$ predicted by each mixture model. To avoid confusion with the GMM used for the proposed channel response estimator, this step will henceforth be referred to as "GMM classifier."

Note that this backend is used for both the proposed method and the benchmarks. In the proposed method, the feature vectors $\mathbf{x}$ correspond to the estimated channel responses, whereas for the benchmarks they are the CQCCs or the DFTspec.

3.5. Figure of merit

In biometric security, performance is commonly evaluated using the equal error rate (EER). It requires the computation of the false negative rate (FNR), the false positive rate (FPR) and a threshold. When the rates are equal, the common value is referred to as the equal error rate. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The EER was the metric used during the ASVspoof 2017 Challenge for performance evaluation and is the adopted evaluation criteria to compare the performance of our proposed system and the benchmarks. The best achieving model is the one that, for a given threshold, provides the lowest EER [15].

4. Experimental Results and Discussion

In this section, we describe our experiments and provide a discussion on the achieved results. Table 1 summarizes our findings for detecting replay attacks on the development set of the ASVspoof 2017 Challenge dataset. Table 2 , in turn, provides results obtained on the evaluation set. For the latter, two training approaches are considered, where i) only the training set of the ASVspoof dataset is used to train the GMM classifier, and ii) where the combined train and development sets are used. These are labeled as columns 'Train' and 'Train + Dev' in the tables, respectively.

Motivated by the findings from [13], we also explore the advantages of applying PCA to the proposed and benchmark methods. As can be seen from the Tables, as expected, adding more data to train the GMM classifiers helped to reduce the EER for all tested methods. Moreover, PCA projections also helped all tested systems. Overall, the proposed method with PCA projection resulted in the lowest EER for both the development and evaluation sets. In the development set, EER went from 9.51 % to 6.87 %, after applying PCA, an improvement of 38 %. For the evaluation set, the benefit of applying PCA on the proposed method was even higher. EER went from 22.08 % to 11.28 % (a gain of 95 %) when using only the training set to train the model, and from 17.66 % to 11.23 %, which represented an improvement of 5 % when using the training and development set to train the model. Similar trend can be observed for the DFTspec. As discussed in Section 3.3, it implies part of the variables of these features are highly correlated and hence can be dismissed. The same assumption can not be made for the CQCC as PCA seems to not help to boost performance. On the contrary, it seems to hurt its results. On reason might be the fact that these features are already decorrelated as they are already operating in a low dimension as they are based on a 90-dimension cepstral coefficients. Overall, the proposed method yielded the best performance in the tested scenarios, showing its potential as a spoofing countermeasure.

Table 1: *Results in terms of EER (%) for replay attack detection on development set of the ASVspoof Challenge 2017.*

Features	EER
CQCC	14.56
DFTspec	14.74
Channel	**9.51**
CQCC (PCA)	11.05
DFTspec (PCA)	9.93
Proposed (PCA)	**6.87**

Table 2: *Results in terms of EER (%) for replay attack detection on evaluation set of the ASVspoof Challenge 2017.*

Features	Train	Train+Dev
CQCC	**21.43**	**15.47**
DFTspec	23.52	17.25
Proposed	22.08	17.66
CQCC (PCA)	21.65	17.12
DFTspec (PCA)	12.86	11.41
Proposed (PCA)	**11.28**	**11.23**

5. Conclusion

In this paper, we propose a new approach for replay attack detection. The proposed method is based on the estimation of the magnitude response of an unknown channel from an observed single-channel speech signal. For that, the log-spectrum average of the clean speech signal is attained using RASTA filtered mel-frequency cesptral coefficients and a Gaussian mixture model. Nuances of acoustic ambience, microphones and playback devices encountered in the spectrum are assumed to be enough information for distinguishing between a bonafide and spoof attack. As such, the proposed method achieved equal error rate (EER) as low as 6.87% on the development set and 11.28% on the evaluation set, thus outperforming two state-of-the-art benchmarks.

6. Acknowledgement

The authors would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fonds de recherche du Québec - Nature et Technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN-2019-05381. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSERC.

7. References

[1] Biometricupdate.com, "Mobile biometric applications," https://www.biometricupdate.com/wp-content/uploads/2017/03/special-report-mobile-biometric-applications.pdf, 2018, [Online; accessed 20-March-2018].

[2] Z. Wu and et al., "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth*

Annual Conference of the International Speech Communication Association, 2015.

[3] T. Kinnunen and et al., "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[4] http://www.asvspoof.org/, "Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," http://www.asvspoof.org/, 2019, [Online; accessed 20-March-2018].

[5] C. Hanilçi, "Features and classifiers for replay spoofing attack detection," in *10th International Conference on Electrical and Electronics Engineering (ELECO), 2017*. IEEE, 2017, pp. 1187–1191.

[6] C. e. a. Wang, "An efficient learning based smartphone playback attack detection using gmm supervector," in *IEEE Second International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 385–389.

[7] J. Gałka, M. Grzywacz, and R. Samborski, "Playback attack detection for text-dependent speaker verification over telephone channels," *Speech Communication*, vol. 67, pp. 143–153, 2015.

[8] N. Gaubitch, M. Brookes, and A. Naylor, "Blind channel magnitude response estimation in speech using spectrum classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2162–2171, 2013.

[9] J. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[10] K. Lee and et al., "The reddots data collection for speaker recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] M. Todisco and et al, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients," vol. 25, 2016, pp. 249–252.

[12] M. Todisco and et al., "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," vol. 45, pp. 516–535, 2017.

[13] M. Alam, G. Bhattacharya, and P. Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 393–398.

[14] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[15] I. C. et al., "Evaluation methodologies for biometric presentation attack detection," in *Handbook of Biometric Anti-Spoofing*. Springer, 2019, pp. 457–480.