Speech Audio Splicing Detection and Localization Exploiting Reverberation Cues

Davide Capoferri, Clara Borrelli, Paolo Bestagini, Fabio Antonacci, Augusto Sarti, Stefano Tubaro
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

***Abstract*—Manipulating speech audio recordings through splicing is a task within everyone's reach. Indeed, it is very easy to collect through social media multiple audio recordings from well-known public figures (e.g., actors, politicians, etc.). These can be cut into smaller excerpts that can be concatenated in order to generate new audio content. As a fake speech from a famous person can be used for fake news spreading and negatively impact on the society, the ability of detecting whether a speech recording has been manipulated is a task of great interest in the forensics community. In this work, we focus on speech audio splicing detection and localization. We leverage the idea that distinct recordings may be acquired in different environments, which are typically characterized by distinctive reverberation cues. Exploiting this property, our method estimates inconsistencies in the reverberation time throughout a speech recording. If reverberation inconsistencies are detected, the audio track is tagged as manipulated and the splicing point time instant is estimated.**

I. Introduction

In the last few years, the massive presence of social media in everybody's life has strongly impacted the way people communicate and news circulate. Unfortunately, one negative aspect of this change has been the increased popularity of fake news spreading. This phenomenon is so diffused, that fake news occasionally break the social media wall and find their way to the mainstream media as well. This has a greatly negative impact on society, as misinformation and hoaxes are often subtly carved to damage people or to obtain some financial or political gain through opinion formation campaigns.

One of the reasons fake news are not always immediately recognized as such is that they often come with videos that make the news look more realistic. Indeed, videos are often considered a strong evidence, as it is common sense to assume that they are hard to be manipulated. However, multiple techniques to realistically edit videos and audio tracks actually exist. These are not always particularly complex to be used.

Considering videos, it is possible to change or swap the face of an actor through face2face [1], neural texture [2], faceswap [3] or deepfake [4] technology. These techniques require some computational power, but provide incredibly realistic results. Considering audio, it is possible to apply similar style transfer methods to turn one voice into another [5], [6]. However, audio speech editing is also possible using cheap yet convincing techniques. This is the case of audio splicing, i.e., composing a new speech by segmenting and concatenating different recordings. After a fake speech and video are produced, they can be put together by applying audio-aware lip synthesis technologies [7], [8], thus becoming a serious threat [9].

Within this context, we consider the problem of speech audio splicing detection and localization. This is, given a speech audio track, to understand whether it comes from a single recording or it is a composition of two separate tracks. If it is a composition, we also estimate the splicing point, i.e., at which sample in time the first original recording ends and the second one starts.

Detecting speech audio splicing is not the same as detecting a fake audio track generated through speech synthesis methods (e.g., style transfer [5], [6], text-to-speech [10], etc.). In this last scenario, fake speech tracks contain some global traces due to the fake audio generative process. These traces can be exploited by forensic detectors to distinguish real recordings from synthetic audio excerpts [11], [12]. Conversely, all samples in a spliced audio track are pristine by definition, as they come from original audio recordings. Therefore, a splicing detector has to rely on different kinds of traces that highlight the change from one recording to another.

As an example, in [13] the authors detect splicing by searching for signal discontinuities that are enhanced through high pass filtering. In [14], the authors focus on noise traces. The rationale is that different recordings may contain different amount of noise, thus noise level estimates can be used to expose splicing. Another interesting approach is proposed in [15]. Here the authors use a blind channel estimator to detect microphone response footprints. Audio tracks showing more than one microphone footprint are detected as spliced. In [16], inconsistencies in Electrical Network Frequency (ENF) traces are used to expose a splicing. As ENF traces are subtle and might be hindered by high noise levels, in [17] the authors propose to use spectral phase analysis to increase noise robustness. More recently, the authors of [18] propose to exploit acoustic channel impulse response and ambient noise as environmental signature for an audio recording. If this signature changes in time, audio splicing is detected.

Similarly to state of the art work, we also make the assumption that spliced audio excerpts may come from different recordings, which can be characterized by different environmental traces. In particular, motivated by [19], we exploit reverberation time as forensic trace. This measures the

WIFS'2020, December, 6-11, 2020, New York, USA. 978-1-7281-9930-6/20/$31.00

degree of reverberation characteristic of an audio signal propagating within an environment. Given a suspect audio track, our method estimates the reverberation time across different temporal windows and searches for possible inconsistencies. If reverberation time suddenly changes from an instant to another, the audio track is detected as spliced.

II. Background and problem statement

A. Reverberation model

Let us consider an indoor environment enclosed by walls. An acoustic source (e.g., a speaker, a loudspeaker, etc.) and a receiver (e.g., a listener, a microphone, etc.) are present in the room. When the source emits an audio signal, the receiver receives multiple delayed and attenuated copies of the signal. Indeed, the microphone is hit by waves propagating directly from the source to the receiver as well as waves reflected by the ground, the walls and other surfaces. The propagation of the signal from the source to the microphone within the environment can be then well approximately by a Linear Time Invariant (LTI) system. Therefore, the signal acquired at the microphone can be modeled as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau, \qquad (1)$$

where $x(t)$ is the source signal, $h(t)$ is the system impulse response known as Room Impulse Response (RIR), and the operator $*$ represents convolution. As the RIR depends on the environment geometry and the source and receiver position, it contains valuable information about the recording setup.

Let us assume that $x(t)$ is a Dirac function, which can be approximated as a short sound impulse emitted by an omnidirectional point source. The recorded $y(t)$ corresponds to $h(t)$, which is typically composed by a series of attenuated and delayed pulses as shown in Fig. 1. A spherical wave propagates from the source in all directions and the wave-front that first reaches the receiver is the one that follows the direct path from the source to the receiver. Therefore, the first pulse of a RIR represents the *direct signal* propagation. This direct signal is followed by weaker components, i.e., waves that have been reflected by the room walls one or multiple times before reaching the receiver. These reflections, called *early reflections*, have lower intensity because of the increased area of the spherical wave-front as time increases and because of the sound-absorbing property of the walls or objects in the room. As the number of reflections increases, the waves continue to travel in all directions until all the energy has been absorbed. The density of these later reflections increases with time, while the intensity decreases. This decaying *reverberation tail* is often perceived by the listener as the room reverberation.

To summarize in a compact fashion great part of the information contained in a RIR about the reverberation properties of the room, a parameter called Reverberation Time (RT) has been proposed and broadly adopted. The RT is defined as the time interval in which the sound pressure level is reduced by a specific range expressed in dB. This range is typically set from 0 dB to 60 dB, in which case RT is also called T_{60}. The higher the T_{60}, the longer the reverberation.

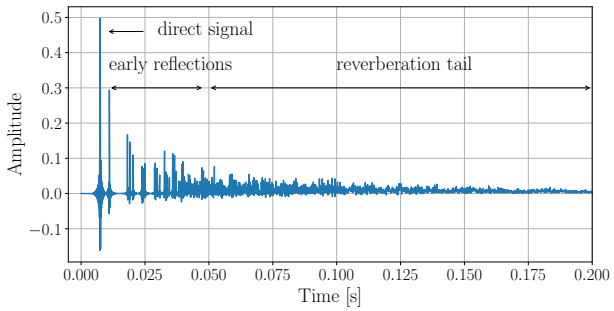


Fig. 1. Components of a typical room impulse response characterizing acoustic propagation from a source to a receiver within a closed environment.

T_{60} can be analytically computed from a RIR. However, when a signal recording is available, estimating the complete RIR is a challenging task. Fortunately, it is possible to estimate the RT directly from an audio recording with some approximations [20]. These methods work particularly well on signals that exhibit small pauses from time to time. This condition is typically fulfilled by speech signals, as no matter how fast a person speaks, some pauses in between words are customarily present. As shall be clear from Section III, we exploit this property in our work.

B. Problem formulation

Formally, let us consider two audio recordings acquired with a single microphone at sampling frequency F_s in two different reverberant environments. These two discrete time signals are defined as

$$\begin{aligned} &y_1(n), \quad n = 0, 1, ..., N_1 - 1, \\ &y_2(n), \quad n = 0, 1, ..., N_2 - 1, \end{aligned} \qquad (2)$$

where N_1 and N_2 are the length of $y_1(n)$ and $y_2(n)$, respectively. A spliced audio track is obtained by concatenating in time $y_1(n)$ and $y_2(n)$, thus it is defined as

$$y_{\text{spliced}}(n) = [y_1(0), ..., y_1(N_1 - 1), y_2(0), ..., y_2(N_2 - 1)].$$

The resulting length of $y_{\text{spliced}}(n)$ is $N_1 + N_2$.

Given a generic audio track, solving the splicing detection problem means understanding whether the audio track is a single recording as $y_1(n)$ or $y_2(n)$, or it is a composition of two recordings as $y_{\text{spliced}}(n)$. If this is the case, solving the splicing localization problem means estimating the splicing time instant, i.e., the sample index $\hat{n}$ where the two sequences $y_1(n)$ and $y_2(n)$ meet ($\hat{n} = N_1$ in this example). In our work we propose a method to solve both problems.

III. Proposed method

The proposed method for speech audio splicing detection and localization verifies the integrity of a suspect signal by analyzing the acoustic properties of the reverberant room in which the recording has been performed. If the reverberation behavior of the environment shows a drastic change within the recording, splicing attack is detected.

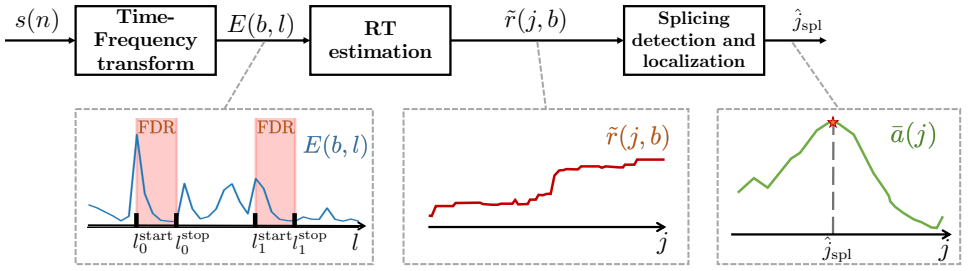


Fig. 2. Pipeline of the proposed method. A signal $s(n)$ is analyzed to estimate RT over time. Splicing is detected and localized through RT inconsistency analysis. The role of each block of the pipeline is explained in the text.

To address this problem, we follow the pipeline depicted in Fig. 2. First, we turn the signal into a time-frequency representation. Then, we estimate the reverberation time on sliding windows. Finally we search for inconsistencies among estimated reverberation times along the recording. In the following, we provide details about each proposed step.

A. Time Frequency transform

The goal of this step is to turn the input signal into a representation that highlights regions useful for RT estimation.

Given a recorded signal $s(n)$ sampled with sampling frequency F_s, we first divide it in J frames $s_j(n)$, $j = 0, 1, ..., J-1$, using a rectangular window of length W with overlap of P samples. The frame length W determines the temporal resolution for RT estimation. Each frame is transformed into a time-frequency representation through Short Time Fourier Transform (STFT), thus obtaining

$$S_j(k,l) = \sum_{n=0}^{M-1} s_j(n) w(n - l(M-V)) e^{-i\frac{2\pi}{M}nk}, \quad (3)$$

where $k = 0, 1, ..., M-1$ is the frequency bin index, $l = 0, 1, ..., L-1$ is the time sample index within the frame, $w(n)$ is a window of length M and V is the overlap between adjacent windows. For the sake of notational simplicity, hereinafter we drop the frame index j whenever not strictly necessary, keeping in mind that the following operations are applied per-frame.

As not all spectral bands are relevant for RT estimation, we adopt an octave band representation of a particular portion of the spectrum. Specifically, we chose B significant octave bands described by their lower (i.e., $f_b^{\min}$, $b = 0, 1, ..., B-1$) and upper (i.e., $f_b^{\max}$, $b = 0, 1, ..., B-1$) frequency limits. Moreover, as phase information is not of interest in our scenario, we compute the energy envelope curve for each band. These two operations lead to

$$E(b,l) = \sum_{k=\lfloor M f_b^{\min}/F_s \rfloor}^{\lfloor M f_b^{\max}/F_s \rfloor} |S(k,l)|^2. \quad (4)$$

An example of $E(b,l)$ for one frame and band is shown in Fig. 2.

B. Reverberation Time estimation

The goal of this step is to estimate a RT for each frame and for each octave-band independently. The adopted algorithm is divided in three steps.

In the first step, we identify and isolate Free Decay Regions (FDRs). These are defined as the portions of the signal where the sound stimulus has already finished and only the reverberation effect is present. These regions can be detected by looking for a persistent energy decrease in time, following the approach introduced in [21]. In a nutshell, the algorithm looks for $E(b,l)$ portions that are monotonically decreasing for at least $\bar{L}$ samples. We therefore obtain a set of I FDRs for each frame and band. Each FDR is described by its start time index l_i^{start}, $i = 0, 1, ..., I-1$ and stop time index l_i^{stop}, $i = 0, 1, ..., I-1$. Two FDRs are shown in the example of Fig. 2 superimposed to the related $E(b,l)$.

In the second step, we apply a modified version of Schroeder's algorithm [22] to each detected FDR to estimate the RT. To this purpose, we compute the energy decay curve, which is the normalized cumulative sum of the energy envelop $E(b,l)$ in dB defined as

$$c_i(b,l) = 10 \log_{10} \left(\frac{\sum_{\lambda=l}^{l_i^{\text{stop}}} E(b,\lambda)}{\sum_{\lambda=l_i^{\text{start}}}^{l_i^{\text{stop}}} E(b,\lambda)} \right), \quad (5)$$

with $l = l_i^{\text{start}}, ..., l_i^{\text{stop}}$. For each band and FDR, we fit a line to $c_i(b,l)$ in the temporal dimension l using a least-square fitting procedure. The slope $d_i(b)$ of the fitted line can be used to estimate the RT value as

$$r_i(b) = \frac{-60/d_i(b)}{F_s/(M-V)}. \quad (6)$$

To obtain a single RT estimate per band, we average the estimated RT $r_i(b)$ using the fitting mean square error $e_i(b)$ as weight, thus obtaining

$$\bar{r}(b) = \frac{\sum_{i=0}^{I-1} e_i(b) r_i(b)}{\sum_{i=0}^{I-1} e_i(b)}. \quad (7)$$

As the process is repeated for each frame, we end up with a RT estimate per frame and band $\bar{r}(j,b)$.

Finally, as RT estimates can be noisy due to the approximation process on short windows, we apply a cleaning operation.

To reduce RT fluctuations over time, we apply a 1D median filter of size R to $\bar{r}(j,b)$, thus obtaining

$$\tilde{r}(j,b) = \text{median}\{\bar{r}(m,b), m \in [j - \lfloor R/2 \rfloor, ..., j + \lfloor R/2 \rfloor]\}.$$

An example of $\tilde{r}(j,b)$ for one band is shown in Fig. 2, where it is possible to see an increase in the estimated RT approximately from the middle of the signal.

C. Splicing detection and localization

The goal of this step is to analyze RT estimates over time and detect and localize an inconsistency, if present.

If audio splicing occurs at time index j_{spl}, we expect that RT changes after j_{spl}. Therefore, $\tilde{r}(j,b)$ values for $j < j_{\text{spl}}$ should be strongly different from $\tilde{r}(j,b)$ values for $j \geq j_{\text{spl}}$ within each band. To check whether this happens, we compare RT estimates before and after each possible j value. If a j providing noticeable RT differences exists, we detect and localize the splicing.

More specifically, for each band, we compute the Absolute Average Difference (AAD) between $\tilde{r}(j,b)$ samples to the left and to the right of each index j. Formally, for the b-th band we compute

$$a(j,b) = \left| \frac{1}{j} \sum_{m=0}^{j-1} \tilde{r}(m,b) - \frac{1}{(J-j)} \sum_{m=j}^{J-1} \tilde{r}(m,b) \right|, \quad (8)$$

for $j = Q, ..., J - Q - 1$, being Q the minimum amount of samples that grants significant statistics before and after the candidate splicing time. To aggregate results over each frequency band, we make use of a weighted average. Formally, we compute the full-band AAD as

$$\bar{a}(j) = \frac{1}{B} \frac{\sum_{b=0}^{B-1} a(j,b) A(b)}{\sum_{b=0}^{B-1} A(b)}, \quad (9)$$

where $A(b)$ is the signal energy in the b-th band. An example of $\bar{a}(j)$ is shown in Fig. 2.

At this point, the full-band AAD $\bar{a}(j)$ should exhibit a pronounced peak in correspondence of the splicing time index, if splicing did occur (as shown in Fig. 2). We therefore search for peaks that have a minimum prominence (10% in our experiments), which measures how much a peak emerges from the neighboring baseline of the signal. If peaks exist, we select the highest one. The position j of this peak is considered the candidate splicing point $\hat{j}_{\text{spl}}$. The height of the peak $\bar{a}(\hat{j}_{\text{spl}})$ is used as splicing detection confidence value. In other words, we detect splicing if $\bar{a}(\hat{j}_{\text{spl}}) > T$, where T is a threshold that can be tuned by observing a small training set of data.

IV. Experimental results

In this section we first present the experimental setup designed for the evaluation step, including the dataset created for the task. Then, we present the metrics and the results for the proposed method compared to some baselines.

A. Dataset

For the evaluation step we have created a dataset which includes both pristine and spliced speech signals affected by reverberations. As already mentioned in Section II-A, a reverberant audio signal can be obtained as the convolution between a dry source signal acquired in an anechoic environment, and a RIR for a specific room and source-receiver position.

As source signals we used part of the ACE dataset [23], which includes 65 utterances from both male and female speakers acquired in an anechoic room with variable length between 15 s and 90 s.

For the RIRs, we decided to include both simulated ones and RIRs acquired in real environments. To create synthetic RIRs we used a Python toolbox called Pyroomacoustics [24], which exploits the Image Source Model algorithm [25] for RIR simulation. We considered 7 shoe box rooms with volumes going from 54 m^3 to 700 m^3 and $\text{T}_{60} \in \{0.31, 0.40, 0.52, 0.62, 0.72, 0.82, 0.93\}$ second. Moreover, for each room two different source-receiver configurations have been considered. This approach allows to quickly create a large set of simulated environments but lacks in describing the diffuse components, due to late reverberation and room irregularities. For this reason, we decided to take into account also real RIRs included in the ACE dataset. These signals are relative to 7 rooms, with volumes varying approximately from 47 m^3 to 360 m^3 and average reverberation time $\text{T}_{60} \in \{0.34, 0.37, 0.39, 0.44, 0.64, 0.65, 1.25\}$ second. Also in this case, two different microphone and source positions have been considered.

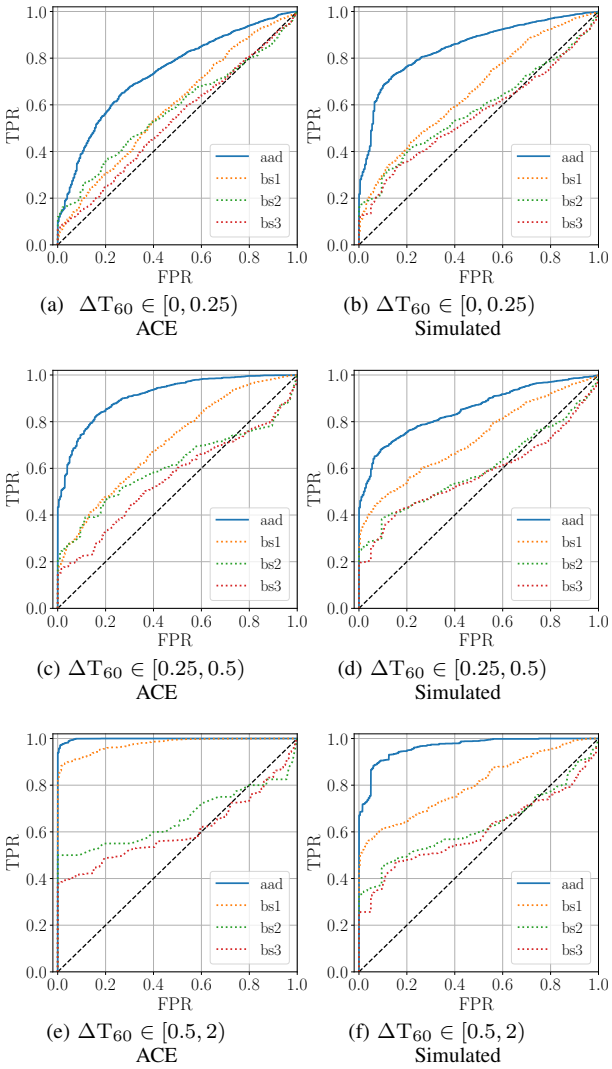
By performing convolution between the considered RIRs and the dry speech signals, we obtained a set of reverberant speech signals, which have been further processed by adding an additive white noise for 3 different Signal-to-Noise Ratio (SNR) levels, namely $\text{SNR} \in \{10 \text{ dB}, 20 \text{ dB}, 30 \text{ dB}\}$.

For the creation of tampered examples, a subset of the resulting speech signals have been concatenated in random position, reproducing the slicing operation. This entire procedure led to a total of approximately 20 000 audio recordings, equally divided in spliced and not spliced instances. Signals convoluted with real and simulated RIRs are always kept apart to allow a separate analysis on the two datasets.

B. Setup

The parameters used for out algorithm are $F_s = 16$ kHz, $W = 32\,000$ samples (i.e., 2 s), $P = \frac{3W}{4}$ samples, $M = 800$ samples (i.e., 0.05 s), $V = \frac{M}{4}$ samples, $B = 6$, $f_b^{\text{min}} \in \{88.4, 176.8, 353.5, 707.1, 1414.2, 2828.2\}$ Hz, $f_b^{\text{max}} \in \{176.8, 353.5, 707.1, 1414.2, 2828.2, 5656.8\}$ Hz, $L = 13$ subframes (i.e., ~ 0.5 s), $R = 7$ subframes (i.e., ~ 0.25 s), $Q = 133$ subframes (i.e., ~ 5 s).

The performance of our method for the detection task is compared to three different baseline methods. They all share the RT estimation step proposed in our method, but they use different indicators to detect whether RT remains constant or changes in time. The first one (*bs1*) makes use of the standard deviation of the estimated RT. The second one (*bs2*) makes



(a) $\Delta T_{60} \in [0, 0.25)$ ACE (b) $\Delta T_{60} \in [0, 0.25)$ Simulated

(c) $\Delta T_{60} \in [0.25, 0.5)$ ACE (d) $\Delta T_{60} \in [0.25, 0.5)$ Simulated

(e) $\Delta T_{60} \in [0.5, 2)$ ACE (f) $\Delta T_{60} \in [0.5, 2)$ Simulated

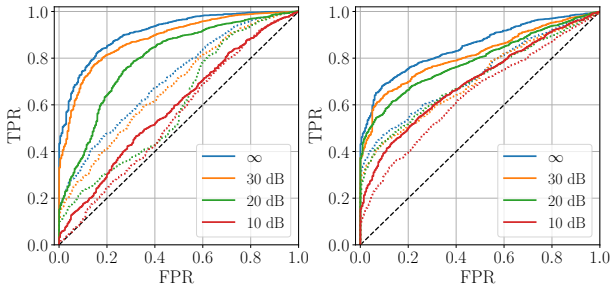
Fig. 3. ROC curves obtained with different ΔT_{60} values compared to all baseline methods. Figures (a), (c) and (e) are relative to the ACE dataset. Figures (b), (d) and (f) are relative the simulated dataset.

use of the difference between the maximum and minimum RT estimates. The third one (*bs3*) makes use of the maximum magnitude of the RT gradient in time. Whenever one of these indicators exceeds a threshold, splicing is detected.

C. Detection results

For the evaluation of the splicing detection task, we adopted ROC curves, which show True Positive Rate (TPR) and False Positive Rate (FPR) pairs for the different threshold values T.

We first present ROC curves in a noiseless scenario for different ΔT_{60}, i.e., the absolute value of the difference between reverberation time before and after the splicing point. The smaller the ΔT_{60}, the closer the RTs before and after the splicing point. Therefore, a small ΔT_{60} depicts a more challenging setup. Fig. 3 reports results for the two different datasets against the baseline methods. We can observe that the higher is ΔT_{60}, the better is the performance of the proposed



(a) ACE dataset (b) Simulated dataset

Fig. 4. ROC curves for different SNR values compared to baseline *bs1* (dashed)

method as expected. The proposed method always outperforms all baselines, confirming that the use of a deeper statistical analysis of reverberation times through AAD enables better performance especially for low ΔT_{60} values. Finally, notice that the achieved performance are better on ACE dataset for high ΔT_{60}, whereas they look better on the simulated dataset for smaller ΔT_{60}. This highlight the impact that diffusive events that are present in ACE but not in the simulated data impact on RT estimation.

To evaluate the impact of additive noise, we also report ROC curves for different SNR values in Fig. 4 . In this case, we set ΔT_{60} to the interval $[0.25, 0.5]$, and only show the best baseline (i.e., *bs1*). Notice that, when the SNR decreases, all methods lose effectiveness in detecting spliced recordings as expected. Nonetheless, for SNR=30 dB detection is still adequate, in particular for the ACE dataset.

From the above analysis, it is possible to select an appropriate threshold value T according to the desired ratio between TPR and FPR.

D. Localization results

Regarding the splicing localization task, a preliminary consideration is necessary. The proposed method relies on RT, which can only be estimated within FDRs. Therefore, we can only tell whether a splicing occurs in-between two different FDRs, but we cannot estimate the precise time instant. As a consequence, the splicing point localization is affected by an intrinsic error, determined by the distance between two successive FDRs. We therefore evaluate splicing localization by providing the correct localization rate defined as the fraction of times we predict the splicing point up to an error of 5 s with respect to the real splicing.

Tables I and II show localization rates obtained for the two dataset and for different values of ΔT_{60} and SNR. Best results are obtained for noiseless recordings and high values of ΔT_{60} as expected. In particular, we get 86% of correct localization on the ACE dataset. As already observed for the detection task, the algorithm tested on the ACE dataset gives better results with respect to the simulated one. This is due to the fact that simulated RIRs are an approximation of a real-world scenario. Nonetheless, with real RIRs we achieve better results.

TABLE I
LOCALIZATION RATES FOR ACE DATASET.

ΔT_{60} \ SNR	10 dB	20 dB	30 dB	∞ dB
[0, 0.25)	0.029	0.030	0.096	0.202
[0.25, 0.5)	0.065	0.231	0.408	0.535
[0.5, 2)	0.606	0.70	0.836	0.861

TABLE II
LOCALIZATION RATES FOR SIMULATED DATASET.

ΔT_{60} \ SNR	10 dB	20 dB	30 dB	∞ dB
[0, 0.25)	0.085	0.182	0.285	0.425
[0.25, 0.5)	0.196	0.417	0.574	0.680
[0.5, 2)	0.257	0.492	0.626	0.744

When the noise component increases or the change in RT values is less accentuated, localization performance degrades. It is interesting to observe that on the ACE dataset the method seems to suffer more from small values of ΔT_{60} than from lower SNR values. We can assume that, when the difference before and after the splicing in RT is noticeable enough, the performance is still positive, despite the low SNR value.

V. CONCLUSIONS

In this paper, we faced the problem of speech audio splicing detection and localization. The goal is to understand whether a speech signal is original or it has been manipulated through splicing. To solve this problem, we proposed a method that exploits inconsistencies in reverberation time. Specifically, we estimate the amount of reverberation in time from an audio signal, and we verify whether reverberation time suddenly changes. The proposed method has been validated on real and simulated room impulse responses applied to male and female speakers with different amount of additive noise.

The proposed method is tailored to speech signals as it requires multiple free decay regions to be present in the recording. Future work will be devoted to more robust reverberation time estimation methods that can be applied also to other kinds of signals. Moreover, an iterative procedure to detect and localize more than one splicing point will be devised.

ACKNOWLEDGMENT

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

REFERENCES

[1] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, pp. 1–12, 2019.

[3] "Faceswap," https://github.com/MarekKowalski/FaceSwap/.

[4] "Deepfakes github," https://github.com/deepfakes/faceswap.

[5] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[6] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[7] L. Chen, Z. Li, R. K Maddox, Z. Duan, and C. Xu, "Lip movements generation at a glance," in *European Conference on Computer Vision (ECCV)*, 2018.

[8] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[10] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017.

[11] A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, and S. Tubaro, ""Hello? Who am I talking to?" A shallow CNN approach for human vs. bot speech classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[12] E. A. AlBadawy, S. Lyu, and H. Farid, "Detecting ai-synthesized speech using bispectral analysis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[13] A. J. Cooper, "Detecting butt-spliced edits in forensic digital audio recordings," in *AES International Conference: Audio Forensics: Practices and Challenges*, 2010.

[14] X. Pan, X. Zhang, and S. Lyu, "Detecting splicing in digital audios using local noise level estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[15] L. Cuccovillo, S. Mann, M. Tagliasacchi, and P. Aichroth, "Audio tampering detection via microphone classification," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.

[16] P. A. A. Esquef, J. A. ApolinÃ¡rio, and L. W. P. Biscainho, "Improved edit detection in speech via enf patterns," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[17] X. Lin and X. Kang, "Exposing speech tampering via spectral phase analysis," *Digital Signal Processing*, vol. 60, pp. 63–74, 2017.

[18] H. Zhao, Y. Chen, R. Wang, and H. Malik, "Audio splicing detection and localization using environmental signature," *Multimedia Tools and Applications*, vol. 76, pp. 13 897–13 927, 2017.

[19] H. Malik, "Acoustic environment identification and its applications to audio forensics," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 8, pp. 1827–1837, 2013.

[20] T. d. M. Prego, A. A. de Lima, R. Zambrano-López, and S. L. Netto, "Blind estimators for reverberation time and direct-to-reverberant energy ratio using subband speech decomposition," in *IEEE workshop on applications of signal processing to audio and acoustics (WASPAA)*, 2015.

[21] J. Vieira, "Automatic estimation of reverberation time," in *Audio Engineering Society Convention*, 2004.

[22] M. R. Schroeder, "New Method of Measuring Reverberation Time," *Journal of the Acoustical Society of America*, vol. 37, pp. 409–412, 1965.

[23] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Estimation of room acoustic parameters: The ACE challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 24, pp. 1681–1693, 2016.

[24] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: a python package for audio room simulation and array processing algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[25] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, 1979.