Adaptive Multi-Trace Carving for Robust Frequency Tracking in Forensic Applications

Qiang Zhu, Student Member, IEEE, Mingliang Chen, Student Member, IEEE, Chau-Wai Wong, Member, IEEE, and Min Wu, Fellow, IEEE

Abstract—Many information forensic challenges often boil down to the problem of frequency extraction. Previous work has studied tracking frequency components by exploiting the temporal correlation. However, limitations of prior frequency extraction approaches, such as limited performance in noisy conditions, inability to track multiple frequency components, or inefficient real-time implementation, restrict their deployment in many challenging information forensic tasks. To address these issues, we propose Adaptive Multi-Trace Carving (AMTC), a unified approach for detecting and tracking one or more subtle frequency components in very low signal-to-noise ratio (SNR) conditions and in pseudo-realtime with low delay. AMTC treats the signal's time-frequency representation as the input and identifies all frequency traces with dominating energy through iterative dynamic programming and adaptive trace compensation. AMTC considers a long duration of high trace energy as the indicator of frequency presence and accurately detects the frequency component. Extensive experiments using both synthetic and real-world forensic data reveal that the proposed method outperforms several representative prior methods under low SNR conditions and can be implemented in near-realtime settings. The effectiveness of the proposed algorithm may empower the development of new frequency-based forensic technologies and other small-signal applications.

Index Terms—spectrogram, multi-trace tracking, dynamic programming, heart rate, electric network frequency (ENF).

I. INTRODUCTION

MANIPULATION of the media content has become much easier than before due to the recent development of advanced editing tools, the availability of vast amounts of training data [2], [3], and the easy access to a number of trained deep models [4], [5]. As a result, important information contained in digital recordings, such as the time of the recording, the place of the recording, and the identity of a person in the scene, can no longer be deemed trustworthy.

To address this problem, recent research has been focused on digital/information forensics that is “concerned with determining the authenticity, processing history, origin of the digital multimedia content with no or minimal reliance on side channels other than the digital content itself” [6]. The authentication process in many such applications is heavily dependent on the imperceptible environmental frequency traces, e.g., Electric Network Frequency [7]–[9], or the physiological frequency traces, e.g., remote photoplethysmogram [10]–[12], in the multimedia recordings. In the presence of a reference source to compare and validate, a forensic investigator might first extract the traces from the recording and then test the authenticity by evaluating the similarities between certain statistics of the reference trace and those of the extracted ones.

As the extraction of frequency traces often plays a key role in the aforementioned forensic applications, one needs to carefully answer the following questions before deploying a frequency estimator:

1) Can the frequency components be detected from the digital recording?
2) If a frequency component is detected, can the frequency be accurately estimated, especially in low signal-to-noise ratio (SNR) conditions?

Solving the above problems can be nontrivial due to the relatively low signal strength of the components-of-interest compared with those of other audio or visual contents in the recordings. To successfully estimate the frequency of interest within the noisy signal, an algorithm must be robust under strong noise and has the capability to exclude strong interference.

In this paper, we exploit the signal’s time-frequency representation, such as the spectrogram, to perform the frequency estimation. We propose a multiple frequency traces tracking and detection method based on iterative dynamic programming and adaptive trace compensation for such media forensic tasks. Inspired by the seam carving algorithm for content-aware image resizing [13], we treat finding a smooth frequency trace as finding the maximum energy trace in a spectrogram, with an additional regularization term that favors close frequency estimates in consecutive time bins. Such a problem is efficiently solved using dynamic programming.

In many forensic applications, the presence of multiple traces within the frequency range of interest is possible. We propose an iterative frequency tracking method named Adaptive Multi-Trace Carving (AMTC) to track all candidate traces. We apply the proposed single frequency tracking method to obtain the dominating frequency. We then compensate the previous trace energy at the end of each iteration to facilitate the estimation of the next trace. After several iterations, all traces within the frequency range of interest will be obtained. In Fig. 1 we show two tracking results using AMTC on highly-corrupted signals. Note that in both cases, AMTC works...
The temporal tracking of signal traces is needed. To other time-frequency visualizations of the signal for which in this paper, while our proposed techniques can be applied almost perfectly, and the estimation result is almost identical sensor.

Fig. 1. (a) Spectrogram of a synthetic $-10$ dB signal with three frequency components (white dashed line) and the frequency estimates using AMTC (blue line). (b) Spectrogram of a remote-photoplethysmogram signal with weak pulse trace embedded in a strong trace induced by the motion of a subject exercising on an elliptical machine [14] and (d) the spectrogram overlaid with pulse rate estimate (blue line) after compensating first trace estimate (magenta line) using AMTC. The estimation result is compared with the heart rate (white dashed line) simultaneously measured by an electrocardiogram based sensor.

almost perfectly, and the estimation result is almost identical with the reference. An efficient quasi-realtime algorithm is also proposed by utilizing the Markovian property of traces and introducing a bidirectional time window. We call it the online-AMTC. Note that we mainly consider the spectrogram in this paper, while our proposed techniques can be applied to other time-frequency visualizations of the signal for which the temporal tracking of signal traces is needed.

The contributions of this work are summarized as follows:

1) For the task of the frequency-based media forensics, we proposed a robust frequency tracking and detection approach which could track multiple frequency traces in a very low (usually $-10$ dB) SNR condition accurately and efficiently. This method works in general for different levels of frequency variation and does not assume the availability of training data to learn prior knowledge of the signal characteristics.

2) We adapt the offline-AMTC algorithm into an efficient near-realtime implementation. We reduce the computational complexity with a queue data structure and maintain the performance compared with the offline version.

3) We conduct extensive experiments using challenging synthetic and real-world forensic data. Several estimation methods initially proposed for other applications (e.g., the pitch estimation) are implemented, re-trained (the factorial hidden Markov model based method [15]), and compared. The results demonstrate that our approach outperforms other existing arts in terms of accuracy and efficiency.

4) We present a novel detection method based on the AMTC framework to accurately test the presence of trace and discuss other considerations using the approach, such as estimation of the number of frequency components and the benefit from human-in-the-loop involvement.

The rest of the paper is organized as follows. In Section II, the background information and the related work about the frequency tracking problems are discussed. In Section III, we formulate the problem of single trace tracking and solve it using dynamic programming. In Section IV-A, we propose the offline multi-trace tracking method or the offline-AMTC, based on an iterative and greedy search strategy. In Section IV-B we present the online-AMTC. In Section V, we show that AMTC outperforms the state-of-the-art methods on both synthetic and real-world data. In Section VI we evaluate the impact to the performance due to various factors. In Section VII, we discuss common problems that can be addressed with AMTC and the limitations of this algorithm. In Section VIII we conclude the paper.

II. BACKGROUND AND RELATED WORKS

A. Micro Signal Extraction Problem

A number of information forensic challenges often boil down to the frequency extraction problem, where “the signal-of-interest has smaller magnitudes, typically one order of magnitude or more, than the dominating signals” [16]. Among all such forensic applications, the Electric Network Frequency (ENF) and the remote-photoplethysmography (rPPG) are two emerging techniques that may be used to determine the media authenticity. We discuss briefly these two applications as part of the background that inspired this work.

ENF signal can be captured by audio recordings made near mains-powered appliances due to electromagnetic interference, acoustic hum, and mechanical vibrations. ENF signal can also be captured by photo-diodes and cameras due to ENF-induced flicking of mains-powered light sources. As the ENF variation at each time and location instant differs from each other, the time and the place of the recording can be validated by matching its ENF with the reference recording obtained from the power.

The rPPG technique is an attractive approach to address face forensic problems [11], [12]. It has been shown in [10] that a person’s instantaneous pulse rate (PR) could be extracted from his/her face video by examining the pulse-induced color change appeared on the facial skin pixels, even when the video contains significant subject motion and environmental illumination change [14], [17]. Similar to the ENF, the characteristics of the PR could be potentially exploited for this physiological forensic task, if a reference PR recording of that person is available [1].

In both forensic applications, some partial overlap of multiple frequency traces within a certain frequency range is expected. For the ENF scenario, strong acoustic interference from other sources may very well dominate over weak ENF traces. For the rPPG scenario, subject motion frequency trace may coexist with the pulse frequency trace, as illustrated

$^1$We envision the coming era of wide availability of such a human physiological recordings for the public health purpose, with the increasing adoption of the wearable devices in this era of Internet of Things and 5G technology.
B. Prior Art on Frequency Tracking

Traditional frequency estimation algorithms are often applied individually to each temporal segment, assuming segment-wise signal stationarity. Subspace methods such as multiple signal classification (MUSIC) [18] and estimation of signal parameters via rotational invariance technique (ESPRIT) [19] build pseudo power spectra using parametric models of pure sinusoids. These frame-wise estimation algorithms cannot explicitly exploit the temporal correlation of neighboring segments and become less accurate as the SNR drops and frequently generate outliers.

The problem of tracking a single frequency component has been extensively studied. In [20], a sequential Monte Carlo method was proposed, and importance sampling was used to approximate the posterior distribution of each frequency state. However, without a backward smoothing procedure, the output tracking results tend to be inaccurate when substantial interference exists, and the resampling stage makes the algorithm time-consuming. In [21], a prior knowledge of trace dynamic was utilized, and the problem was formulated as a hidden Markov model (HMM) problem. The maximum a posteriori probability estimate was efficiently calculated by running a Viterbi solver. However, HMM requires both the modeling and calibration of a key building block, the emission probability. Such a pre-calibration requirement often makes this method hard to be deployed in real-world forensic tasks, especially when the training data is unavailable. The recently developed Yet Another Algorithm for Pitch Tracking (YAAP) [22] focused on single pitch estimation of speech signal based on both spectrogram and correlogram. The authors proposed using dynamic programming to estimate the fundamental frequency trace from a set of candidate peaks of proposed harmonic spectral features. A similar tracking method can be found in [23]. Such local-peak based methods guarantee excellent performance in high SNR cases, but often generate biased estimates under low SNR, as the probability that a local peak represents the actual signal frequency drops significantly.

The problem of tracking multiple frequency components from the spectrogram image has also been investigated. Image processing techniques such as morphological operators [24], active contour [25] methods have been applied to this area, but these methods may be difficult to be adapted to realtime tracking algorithms. Wohlmutz et al. [15] modeled the probability of pitch using Gaussian mixture models (GMMs), and then used the junction tree algorithm to decode a speaker-dependent factorial HMM (fHMM). A similar approach can be found in [26], where the emission probability was modeled by a deep neural network (DNN). Although both methods provide excellent performance in terms of accuracy, it is sometimes impossible to fit into real-world needs for the following two reasons. First, the training phase requires a large amount of real-world data, which is often unavailable for most tasks. Second, it is relatively time-consuming to compute the frame-wise joint emission probability and to decode the fHMM with the junction tree algorithm. More recent studies [27], [28] proposed to use linear programming to find the best connection path of the frequency peaks on the spectrogram. These two methods first obtain all frequency peaks in the spectrogram as candidates and then find the best path from the candidates via linear programming. For low SNR scenarios, such approaches may find a large number of frequency peaks as the candidates, leading to huge memory and computational cost that is not scalable.

III. TRACK A SINGLE FREQUENCY TRACE

In this section, we present a trace tracking method which provides a practical and robust solution for tracking a single frequency trace. We first discuss the problem formulation by taking into account the energy as well as the smoothness of the trace. We adopt a dynamic programming algorithm to efficiently search for a candidate of the optimized trace. We then solve the trace detection problem by thresholding test statistic of the candidate trace.

A. Problem Formulation

We first formulate a frequency tracking problem for the scenarios that only a single trace exists in a frequency range of interest. Let \( Z \in \mathbb{R}_+^{M \times N} \) be a magnitude of a signal spectrogram image, which has \( N \) discretized bins along the time axis and \( M \) bins along the frequency axis. We define a frequency trace as

\[
\mathbf{f} = \{(f(n), n)\}_{n=1}^N,
\]

where \( f : [1, N] \to [1, M] \) is a function. Given the spectrogram \( Z \) and a candidate trace \( \mathbf{f} \), we define an energy function for the trace as \( E(\mathbf{f}) = \sum_{n=1}^{N} Z(f(n), n) \). A reasonable estimate of the frequency trace for the given signal is the trace \( \hat{\mathbf{f}} \) that maximizes the energy function shown as follows

\[
\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} E(\mathbf{f}).
\]

Problem (2) is equivalent to the peak finding method [7], [29] where \( \hat{f}(n) = \arg\max_{f(n)} Z(f(n), n), \forall n \in [1, N] \). It also shares similar spirit as the weighted average approach [7].

To take into consideration the smoothness assumption of the trace along the time, we add a regularization term that penalizes jumps in the frequency value. We model the change of the frequency value between two consecutive bins at \( n - 1 \) and \( n \) as a one step discrete-time Markov chain, characterized by the prior distribution function \( P_m \) and the transition probability matrix \( P \in \mathbb{R}^{M \times M} \), where \( P_m = \mathbb{P}(f(1) = m) \) and \( P_{m'm} = \mathbb{P}(f(n) = m | f(n-1) = m') \), \( \forall m, m' = 1, \ldots, M \), and \( \forall n = 2, \ldots, N \). Note that we assume \( P_m \) to be uniformly distributed throughout this paper to treat the initial presence of each frequency state equally. The regularized single trace frequency tracking problem is formulated as follows

\[
\end{align*}

in Fig. 1(b). The use of a multiple trace searching strategy increases the chance to find the correct trace-of-interest and thus reduces the false positive rate. The real-time compatibility of a frequency tracking algorithm can also be important for the rPPG scenario when the video data is in streaming mode and the authentication task is time-sensitive.
transitions, i.e., the total energy to $-\infty$ by setting $P$. Note that we can avoid transitions from state $\hat{\delta}$ for $\hat{f}$.

Second, we find the optimal solution by backtracking from $\hat{f}(n)$ column by column for all entries $(m,n)$ as follows

$$G(m,n) = \begin{cases} Z(m,n) + \lambda \log P_m, & n = 1; \\ Z(m,n) + \max_{m'} \{ G(m',n-1) + \lambda \log P_{m'm} \}, & n > 1 \end{cases}$$

After completing the calculation at column $n = N$, the maximum value of the $N$th column is denoted as $\hat{f}(N)$. Second, we find the optimal solution by backtracking from the maximum entry of the last column of the accumulated map $G$. Specifically, we iterate $n$ from $N - 1$ to 1 to solve for $\hat{f}(n)$ as follows

$$\hat{f}(n) = \arg \max_{f(n)} \{ G(f(n),n) + \lambda \log P_{f(n)f(n+1)} \}.$$ (5)

Note that we can avoid transitions from state $m'$ to state $m$ by setting $P_{m'm} = 0$, as the regularized term would penalize the total energy to $-\infty$. If we assume uniform random walk transitions, i.e., $P_{m'm} = \frac{1}{2k+1}$, $|m'-m| \leq k$, then problem (3) is degenerated to the seam carving problem defined in [13], and in this case the value $\lambda$ does not affect the solution.

C. Trace Existence Detection for a Given Time Window

To determine the existence of a frequency component in a specific time interval, we first make independent decisions for every frame within the time interval on the existence of the frequency component and then refine the decisions by taking neighborhood correlations into consideration. We refer to those frames with a frequency component as voiced frames, or otherwise as unvoiced frames. We propose to test the existence of a frequency component by evaluating the relative energy of the detected trace. A test statistic named the Relative Energy Ratio (RER) is defined as follows:

$$\text{RER}(n) = \frac{|F(n)| \cdot Z(\hat{f}(n),n)}{\sum_{m \in F(n)} Z(m,n)}.$$ (6)

where $F(n) \triangleq [1,M] \max \{ 1, |\hat{f}(n) - \delta_f|, \min(1, |\hat{f}(n) + \delta_f|) \}$ is a conservative set of frequency indices that does not contain the frequency indices around the estimated frequency; $\delta_f$ is a predetermined parameter, and $|\cdot|$ is the cardinality of a set. It is evident that the higher RER($n$) is, the more probable that $n$th frame is voiced. The decision is made by comparing the test statistic RER($n$) with an empirically determined threshold $\Delta_{\text{RER}}$. A discussion about the optimal selection of $\Delta_{\text{RER}}$ will be presented later in Section V-A3.

Once the length of the shortest possible unvoiced segment and voiced segment, i.e., $\Delta_1$ and $\Delta_2$, are available, a post-processing to smooth the initial detection result could further improve the detection accuracy. Specifically, we propose to group consecutive unvoiced frames into a segment when the length is greater than $\Delta_1$, and then group consecutive voiced segments into one segment if the distance between the two is smaller than $\Delta_2$. Figs. 2(e) and (f) illustrate two such decision making processes. Note that the final decisions can exclude all
short segments, and the result is more robust compared to that of the initial decision.

IV. TRACK MULTIPLE TRACES VIA ITERATIVE FREQUENCY COMPENSATION

In the previous section, we have introduced a single frequency trace tracking and detection method using dynamic programming and trace existence testing, respectively. For some forensic tasks such as extracting pulse rate from the face video containing subject’s motion, as shown in Fig. 1(c), some forensic tasks such as extracting pulse rate from the face video containing subject’s motion, as shown in Fig. 1(c), interest is possible, and the dominating trace in the spectrogram might not be the one of interest. A crude deployment of any single trace tracking method on such tasks would generate completely wrong answers. To address this problem, we strategically extend the single trace tracking method to be able to track multiple traces by extracting trace iteratively to find all candidates. We name this method the Adaptive Multi-Trace Carving (AMTC). In the rest part of this section, we first present the offline version of AMTC (offline-AMTC), when the trace estimate is optimized according to the entire available signal. We next adapt the offline-AMTC to an efficient online version (online-AMTC), which runs in quasi-realtime with low delay.

A. Offline-AMTC

Similar to the iterative nature of the seam carving algorithm [13], multiple traces can be greedily searched for by iteratively running the single trace tracker proposed in Section III. However, as frequency energy is diffused around the center of each trace due to the windowing effect and violation of the signal stationary assumption, multiple consecutive trace estimates may reside in the same diffused spectral region belonging to a single frequency component without compensating the diffused spectral energy.

To solve this problem, we attenuate the diffused energy around the estimated frequency trace at the end of each iteration once we obtain the estimated frequency trace. Specifically, suppose \( \hat{f}(t) \) is the estimated frequency trace at the \( t \)th iteration. For each time frame of the spectrogram, i.e., \( Z(t)(1 : M, n) \), we search for a left boundary point \( m_{1(t)}(n) \) from \( f(t) \) to its left side. We set \( m_{2(t)}(n) = m \), if \( m \) is the first point that is either a local minimum point in \( Z(t)(1 : M, n) \) or a local minimum point in the first-order difference of \( Z(t)(1 : M, n) \). The search of the right boundary point \( m_{2(t)}(n) \) works similarly except it considers the local maximum point in the first-order difference of \( Z(t)(1 : M, n) \). In this paper, we call \( Z(t)(m_{1(t)}(n) : m_{2(t)}(n), n) \) the energy bump of \( \hat{f}(t) \).

One example of the trace compensation process is shown in Fig. 3. The plot in (d) shows the spectral energy distribution centered at \( f(t) \), which corresponds to the light blue vertical line in (b). In this case, \( m_{1(t)}(n) \) is selected as the first local minimum point, and \( m_{2(t)}(n) \) as the local maximum point in the first-order difference of \( Z(t)(1 : M, n) \). Based on \( m_{1(t)}(n) \) and \( m_{2(t)}(n) \), we propose to use a reverse Gaussian-shaped function to compensate the energy of the estimated frequency component. The updated equation for the compensated power spectrum at the \((l+1)\)st iteration is as follows

\[
Z(t+1)(m, n) = \left[ 1 - \exp \left( -\frac{(m - \hat{f}(t)(n))^2}{2\hat{\sigma}^2 (t)(n)} \right) \right] \cdot Z(t)(m, n),
\]

where \( \hat{\sigma}^2 (t)(n) = \frac{\sum_{m=m_{1(t)}(n)}^{m_{2(t)}(n)} \cdot \sum_{n=n_{1(t)}(n)}^{n_{2(t)}(n)} Z(t)(m, n)}{n_{2(t)}(n) - n_{1(t)}(n)} \) is used to quantify the width of the energy bump at the \( l \)th iteration. The pseudo code of the offline-AMTC is shown in Algorithm 1.

In Fig. 2, we give an example of two-trace estimation process on a synthetic heart beat signal. The final estimate is almost

---

**Algorithm 1 Offline Adaptive Multi-Trace Carving (offline-AMTC)**

1: procedure AMTC\((Z, L) \rightarrow L\) \(\triangleright\) number of output traces
2: \(Z(1) \leftarrow Z\)
3: \(\hat{f}(1) \leftarrow \arg \max_f E_Z(f) + \lambda P(f)\)
4: \(\hat{v}(1) \leftarrow \text{DetectExistence}(Z(1), \hat{f}(1), \Delta_{RER}, \Delta_1, \Delta_2)\)
5: for \(l \leftarrow 2\) to \(L\) do
6: \(\hat{Z}(l) \leftarrow \text{detect} \quad \hat{f}(l) \leftarrow \arg \max_f E_Z(f) + \lambda P(f)\)
7: \(\hat{v}(l) \leftarrow \text{DetectExistence}(Z(l), \hat{f}(l), \Delta_{RER}, \Delta_1, \Delta_2)\)
8: end for
9: return \(\hat{f}(1:L), \hat{v}(1:L)\)

---

2DetectExistence\((\cdot)\) refers to the trace existence detection algorithm described in Section III-C. \(\hat{v}(l) \in \{0, 1\}^N\) is the trace existence decision with 0 as unvoiced and 1 as voiced.
memory requirement are greatly reduced and are independent
of the forward update rule of \( G \) in (4), it is clear that \( G(n) \) would remain unchanged compared to the output in the previous time instant \( n - 1 \). We therefore only need to update the right most frame \( G(n + k_2) \) and the innovation frame \( Z(n + k_2) \) as shown in the middle box of the first row of Fig 4. \( \hat{f}(1)(n) \) is then obtained via backtracking from \( G_1(n + k_2) \) according to (5). Now, we define the previous backtracking result at time \( n - 1 \) as \( f_{pre}^{(n - 1)}(n - 1 + n + k_2 - 1) \).

During the backtracking process for \( \hat{f}(1)(n) \), if \( \hat{f}(1) = \hat{f}^{pre} \) at the time instant \( T_e \in [n, n + k_2) \), we have \( \hat{f}(1) : T_e = f_{pre}^{(n - 1)}(n : T_e) \). This claim holds because \( G(n) \) remains the same during the process. In this regard, we consider storing and updating \( f_{pre}^{(n - 1)}(n - 1 + n + k_2 - 1) \) in a buffer, whereby the update process of \( f_{pre}^{(1)} \) stops at the instant \( T_e \) if \( \hat{f}(1)(T_e) = f_{pre}^{(1)}(T_e) \), as shown in the right box of the first row of Fig 4. In this way, the computation complexity is further reduced.

Different from the estimation process for the first trace, any change from previous trace estimation \( \hat{f}(1)(d - 1) \) would have influence on the formation of \( Z(1) \), \( G(1) \), and therefore \( \hat{f}(1) \). In order to obtain a robust estimate for \( \hat{f}(l) \), \( l > 1 \), we introduce a look-back length, \( k_1 > 0 \) in this process. As demonstrated from second and third rows in Fig 4, for \( l \)th trace estimation at time instant \( n \), we utilize the previous trace estimates \( \hat{f}(l - 1)(n - k_1 : n + k_2) \) and \( Z(l - 1)(n - k_1 : n + k_2) \) to obtain new \( Z(l)(n - k_1 : n + k_2) \) and \( G(l)(n - k_1 : n + k_2) \), and thus \( \hat{f}(l)(n - k_1 : n + k_2) \). Efficient backtracking can also be achieved using the previous backtracking result, same as the case in estimating the first trace. The details of the online-
AMTC algorithm at \( n \)th iteration is shown in Algorithm 2.

The worst-case computational complexity for the online-
AMTC is \( O(N(k_1 + k_2)LM^2) \), which appears to be \( (k_1 + k_2) \) times slower than the offline version. In the statistical sense, we argue that the expected complexity of the online-
AMTC is much less than the worst-case analysis result because the probability that an entire trace estimate changed from the previous one is low at each time instant. To demonstrate this, we will compare the average computation time running the offline- and online-AMTC in Section |V|A2|

V. PERFORMANCE ANALYSIS OF AMTC

A. Simulation Results and Comparison with Known Ground Truth

1) Single Trace: We first evaluated the performance of the AMTC algorithm using simulated data. For each test signal, we generated a time-varying pulse rate trace present from the beginning to the end of the timeline. More specifically, denote \( s[n] \) as the temporal measurement of the corrupted frequency signal, \( s[n] = \sin(\Phi[n]) + \epsilon[n] \), where \( \Phi[n] = \Phi[n - 1] + 2\pi f[n]/f_s \), \( f[n] \) is the time-varying synthesis frequency, \( f_s \) is

![Flowchart](image)

Fig. 4. A flowchart for the online-AMTC algorithm for three traces estimation process at 1th iteration. (·) above arrows indicates the index of the equation being used.

identical with the ground truth, and the unvoiced segments are

If we define \( L \) as the number of traces to track, the computational complexity for the offline-AMTC is \( O(NLM^2) \). To compare, the fHMM methods [15], [26] requires \( O(NLM^{L+1}) \) without considering operations for computing emission probability. The efficiency of offline-AMTC is mostly explained by the idea of the introduced iterative search. We will later show in Section[V] that the demonstrated efficiency is not achieved at the expense of performance drop.

B. Online-AMTC with Low Delay

The offline-AMTC algorithm minimizes the adverse effect of noise by making use of full-length signals. In a delay-
sensitive scenario that a fixed-length delay \( k \) is allowed, the tracking objective at the time instant \( n \) is to estimate \( \hat{f}(1)(n) \) based on the available spectrogram information \( Z(1) : n + k_1 \). A simple approach runs the offline-AMTC from the first instant 1 to \( n + k_1 \) at each time instant \( n \), costing \( O(nLM^2) \) in time. If the total length of the frame is \( N \), this approach requires \( O(NM) \) in space and \( \sum_{n=1}^{N} O(nLM^2) = O(N^2LM^2) \) in time. Space and time complexities increase quadratically and linearly in \( N \), respectively. This increasing trend of workload will eventually lead to either memory overflow or CPU overload, especially when we expect to run the system in days or even months.

An efficient, quasi-realtime algorithm called the online-
AMTC is developed to address the storage and computational issues mentioned above. We propose to use a fixed-length queue buffer for storing and updating the intermediate result of \( Z(1 : L), G(1 : L), \) and \( \hat{f}(1 : L) \). As a result, the running time and the memory requirement are greatly reduced and are independent of time \( n \).

We introduce the algorithm by first discussing online it-
erations for the estimation process of the first trace. The processing flow of the online-AMTC algorithm at the instant

3For concise representation, we use \( G(n_1 : n_2) \) and \( Z(n_1 : n_2) \) as a shorthand for \( G(1 : M, n_1 : n_2) \) and \( Z(1 : M, n_1 : n_2) \), respectively.
the sampling rate set to 30 Hz, and $\epsilon[n]$ is the noise quantified by a zero-mean white Gaussian process. The variance of $\epsilon[n]$ is an adjustable parameter for achieving different SNR levels. To generate frequency signals $f[n]$ that behave similarly as real-world pulse rate signals, we trained a 9-tap autoregressive model using heart rate signals collected by a Polar H7 chest belt in both exercise mode and still mode. We use beat per minute (bpm) as the frequency unit. The duration of each test signal was 3 minutes. The spectrograms were generated by short-time Fourier transform (STFT) with window length 10 secs and 98% overlap between adjacent frames. We padded zeros to the end of each frame to make neighboring frequency bins 0.17 bpm apart.

We then compared our algorithm with the Particle filter method [20] and the local peak based YAAPT method [22] using a large scale synthetic dataset. The number of particles was set to 1024. We generated 500 trials under each of the five SNR conditions, or 250 for each mode (namely, the exercise and the still cases) using the estimated parameters of the autoregressive models. We used three metrics. Namely, the root mean squared error (RMSE), the error rate (ERATE), and the error count (E_COUNT) defined as follows to evaluate the performance:

- **RMSE** = $\sqrt{\frac{1}{T} \sum_{t=1}^{T} (\hat{f}_t - f_t)^2}$,
- **ERATE** = $\frac{1}{T} \sum_{t=1}^{T} \left| \frac{\hat{f}_t - f_t}{f_t} \right|$, 
- **E_COUNT** = $|\{t \in [1, T] : \left| \frac{\hat{f}_t - f_t}{f_t} \right| > \tau \}| / T$.

where $|\{\cdot\}|$ denotes the cardinality of a countable set, $\hat{f}_t$ and $f_t$ are the frequency estimate and the ground-truth frequency at $t$th time frame respectively, and $\tau$ was chosen to be 0.03 empirically determined from the spread of the frequency components. Fig. 5 shows tracking results of a $-10$ dB synthetic signal with one frequency component using AMTC, YAAPT, particle filter, respectively. In this example, AMTC outputs the best trace estimate among the three without much deviation from the ground truth. The results of overall performance are shown in the first row of Fig. 6 in terms of box plots that each box compactly shows the median, upper, and lower quantiles, and the max and min values of a dataset. It is evident from the box plots that, under all SNR levels, AMTC generally outperforms the particle filter method and the YAAPT not only in terms of the average but also in the variance of the error statistics.

Next, we tested the online-AMTC algorithm using different look-ahead time lengths. The evaluation was conducted using the same setting mentioned above, and the averaged behavior of each look-ahead length is plotted in the second row of Fig. 6. The numbers in the legends indicate the lengths of look-ahead (L.A.) window lengths represented by the number of time bins in the spectrogram. We have two observations from these plots. First, a performance jump from no-look-ahead versus 100-bin look-ahead length is observed, but the performance
with fHMM, we adopt the performance measure proposed in [30] with a slight change. We give details on our experiment setup as well as the error measure below.

To test both algorithms, we generated a corrupted frequency signal \( s[n] \) with two frequency traces, i.e., \( s[n] = \sum_{i=1}^{2} \sin(\Phi_{i}[n] + \epsilon[n]) \). The variance of \( \epsilon[n] \) is tuned to achieve six SNR levels from 0 to \(-10\) dB. To cope with the high computational cost associated with running fHMM at a full scale, we cut signals to 1 minute, set the number of frequency bins to 64, and made neighboring frequency bins 1 bpm apart. The cardinality of frequency state is set to 169 so that it uniformly covers the whole frequency range of interest. For each trace, we also introduced a 20 seconds unvoiced segment.

We estimate the GMM parameters of the fHMM framework using the EM algorithm [31]. For each SNR level, we generated 6000 spectrum frames with a single frequency component for each 169 frequency states (where the first state encodes unvoiced decision). We set the maximum number of components per GMM to 20 and used MDL [15] to determine the number of components automatically. The parameters were trained in an SNR-dependent (SD) and an SNR-independent (SI) fashion (i.e., each SD model was trained only with samples of the corresponding SNR, and the SI model was trained with all samples). We adopted the mixture-maximization interaction model proposed in [15], and set the prior distribution for both fHMM and AMTC uniformly as \( P(f(0) = n) = 1/169, \forall n \), and the transition probability follows a uniform distribution with width parameter \( k = 2 \). Moreover, the voiced to unvoiced transition probability for fHMM was empirically selected as \( P(\text{voiced}|\text{unvoiced}) = 0.2 \), and \( P(\text{unvoiced}|\text{voiced}) = 0.1 \).

To compare the tracking performance, we use the well-adopted error measure proposed in [30] as described below:

- \( E_{ij} \): the percentage of time frames where \( i \) frequency components are misclassified as \( j \).
- \( E_{\text{Gross}} \): the percentage of frames where \( \exists l, \text{s.t.} \Delta f_l > 20\% \). We define the relative frequency deviation \( \Delta f_l = \min l \frac{|f_i - f_{i0}|}{f_{i0}} \), and \( f_{i0} \) is the reference frequency for \( l \)th component.
- \( E_{\text{line}} \): the average relative frequency deviation from the reference of the \( l \)th frequency component for those frames where \( \forall l, \Delta f_l \leq 20\% \).

Note that both \( E_{ij} \) and \( E_{\text{Gross}} \) represent a frame counting measure. We therefore group them together to form the total gross error: \( E_{\text{Total}} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + E_{\text{Gross}} \), and define \( E_{\text{line}} = E_{\text{line}} + E_{\text{line}} \).

To test the performance, we generated 30 tested signal for each SNR level using the same setting mentioned above. We compared the performance of SD-fHMM, SI-fHMM, offline-AMTC and online-AMTC using the aforementioned error measures and the results are listed in Table 1. We depict the distribution of \( E_{\text{Total}} \) and \( E_{\text{line}} \) specifically in Fig. 8. All methods have a similar performance in terms of the fine detection error \( E_{\text{line}} \), while AMTC slightly outperforms fHMM in terms of \( E_{\text{Total}} \), the main contributors of which are \( E_{12} \) and \( E_{21} \). Table 1 shows the average computation time for the \textit{mixmax} likelihood estimation procedure [15].
Fig. 6. First row: Comparison of the performance of single trace tracking by the proposed offline-AMTC, particle filter, and YAAPT methods at different levels of SNR. Statistics of the RMSE, the ERate, and ECount of frequency estimates are summarized using box plots. Second row: Trace tracking performance by the online-AMTC with different levels of look-ahead window length and SNR. The results for the offline-AMTC are also shown in the plots for the comparison purpose.

Fig. 7. (a) Spectrogram of one test instance with SNR = −8 dB; (d) same spectrogram overlaid by ground truth traces. Tracking results by SD-fHMM (b), SI-fHMM (e), offline-AMTC (c), and online-AMTC (f).
### TABLE I

**AVERAGE PERFORMANCE OF FHHMM AND AMTC ON MULTI-TRACE TRACKING TEST**

|               | $E_{01}$ | $E_{02}$ | $E_{10}$ | $E_{12}$ | $E_{20}$ | $E_{21}$ | $E_{\text{Gross}}$ | $E_{\text{Total}}$ | $E_{\text{fine}}$ |
|---------------|----------|----------|----------|----------|----------|----------|-------------------|-------------------|------------------|
| SD-FHMM       | 4.14%    | 1.62%    | 0.36%    | 15.39%   | 0.28%    | 1.78%    | 0.02%             | 23.59%            | 1.79%            |
| SI-FHMM       | 3.48%    | 1.52%    | 0.61%    | 14.37%   | 0.29%    | 2.38%    | 0.03%             | 22.68%            | 1.82%            |
| offline-AMTC  | 1.77%    | 0.28%    | 3.57%    | 2.16%    | 0.45%    | 9.99%    | 0.03%             | 18.27%            | 1.76%            |
| online-AMTC   | 1.75%    | 0.38%    | 3.17%    | 2.65%    | 0.48%    | 8.41%    | 0.03%             | 16.87%            | 1.80%            |

---

#### TABLE II

**AVERAGE COMPUTATION TIME IN SECONDS PER 100 FRAMES**

|               | mixmax likelihood (sec) | Tracking (sec) |
|---------------|-------------------------|----------------|
| SD-FHMM       | 39.47                   | 3.96           |
| SI-FHMM       | 195.86                  | 4.30           |
| offline-AMTC  | N/A                     | 0.10           |
| online-AMTC   | N/A                     | 0.44           |

---

Fig. 8. Box plots of $E_{\text{fine}}$ (top) and $E_{\text{Total}}$ (bottom) of two traces tracking using SD-FHMM, SI-FHMM, online-AMTC, and offline-AMTC on different levels of SNR.

---

### Fig. 9

(a) Ground truth frequency traces at $-10$ dB in spectrogram of a synthetic signal. (b) Three trace estimates by AMTC.

---

### Fig. 10

(a) ROC curve of the proposed trace detection method in different SNR conditions. (b) The zoomed-in plot of the shaded area in (a) with optimal operating points (black circle) and operating points using fixed threshold ($\Delta_{\text{RER}} = 2.41$, pink plus sign).
the post-processing operation using $\Delta_1$ and $\Delta_2$ is summarized using the Receiver-Operating Characteristic (ROC) plot in Fig. 10(a). From the plot, we observe highly accurate detection result in each SNR condition with the Area Under the Curve (AUC) higher than 0.9.

In Fig. 10(b), we show the zoomed-in plot of the shaded area in Fig. 10(a). The optimal operating points in terms of minimizing the sum of false negative and false positive rate are shown in black circles. The operating point corresponding to a fixed threshold, namely, $\Delta_{RER} = 2.41$ (the value we used for the experiments in the paper), are also shown using pink plus signs. Note that the detection results using a fixed threshold value are close to those with the optimal choice at every SNR level, demonstrating the insensitivity of the threshold parameter $\Delta_{RER}$.

### B. Experimental Results on rPPG Data

We evaluated the performance of the proposed method on a real-world dataset from the problem of the pulse rate estimation from facial videos. We show by experiment that AMTC can successfully extract the subtle pulse trace even when the trace is dominated by another frequency component. To test the robustness of the algorithm in a challenging situation, we use the dataset where the video contains significant subject motion [14]. In total, the dataset contains 20 videos in which 10 contain human motions on an elliptical machine, and the other 10 contain motions on a treadmill. Each video is about 3 minutes long in order to cover various stages of fitness exercise. Each video was captured in front of the subject’s face by a commodity mobile camera (iPhone 6s) affixed on a tripod or held by the hands of a person other than the subject. The heart rate of the test subject was simultaneously monitored by an electrocardiogram (ECG)-based chest belt (Polar H7) for reference. The spectrogram of the preprocessed face color feature was estimated using the same set of parameters as in Section V-A.

Fig. 11 gives an example of the tracking result using AMTC with a uniform Markov transition probability model with $k = 60$ for first motion-induced trace estimate and with $k = 2$ for second pulse-induced trace estimate. More freedom of trace dynamic ($k = 60$) was assigned to the first estimate as the variation of motion frequency can be much greater than the heart rate. We noticed for each spectrogram, the traces induced by subject motions dominate over the heart rate trace. Compared to the particle filter-based method that utilizes additional information to compensate for the motion trace [14], AMTC can faithfully track the dominating motion trace and recognize the PR trace as the second trace. Notice that the trace estimate from the particle filter would occasionally deviate to the vertical motion trace. We summarize the mean $\mu$ and standard deviation $\sigma$ of the error measures for all of our videos, and the results are listed in Table III. The average error for AMTC is 2.21 bpm in offline mode and 2.78 bpm in online mode in RMSE and 3.16% in offline mode and 4.01% in online mode in relative error. The performance of AMTC is more than twice as good against the state-of-the-art motion notching + particle filter.

### C. Experimental Results on ENF Data

In this subsection, we tested the performance of the proposed algorithm on a real-world ENF dataset. In total, 27 pairs of one-hour power grid signal and audio signal from a variety of locations in North America were collected and tested. Each pair of signals were simultaneously recorded using a battery-
TABLE IV

|                  | RMSE (Hz) | Pearson’s $\rho$ |
|------------------|-----------|------------------|
|                  | $\hat{\mu}$ | $\sigma$ | $\hat{\mu}$ | $\sigma$ |
| QI                | 0.24      | 0.18            | 0.18         | 0.26      |
| Particle Filter   | 0.04      | 0.07            | 0.55         | 0.37      |
| YAAPT             | 0.16      | 0.12            | 0.23         | 0.28      |
| offline-AMTC      | 0.01      | 0.01            | 0.85         | 0.18      |
| online-AMTC       | 0.03      | 0.02            | 0.81         | 0.20      |

powered Olympus Voice Recorder WS-700M at a sampling rate of 44.1 kHz in MP3 format at 256 kbps. We recorded the reference ENF signal from the power mains of the electrical supply. To limit the voltage to the safe range of the input of a sound card or a digital recorder, we used a step-down transformer to convert the power supply voltage level to 5 V and then used a voltage divider with resistors of 33 Ohm and 33 kOhm to obtain an input of 5 mV [7].

We downsampled the signals to 1 kHz to reduce the computational load, and applied harmonic combining method [32] to obtain robust frequency strips around the nominal frequency, i.e., 60 Hz in North America. The harmonic combining method exploits different ENF components appearing in a signal, and adaptively combines them based on the local SNR to achieve a more robust and accurate estimate than that by using only one component. We obtained the ground truth from the corresponding power grid signals using Quadratic Interpolation (QI) [33], as the SNR is high and frame-wise highest peak method is proved to be the maximum likelihood estimator of signal frequency [29]. We use RMSE and Pearson correlation coefficient $\rho$ of the estimated versus the ground-truth sequence of frequency variations as two performance indices. They are two well-adopted error measures for ENF estimation.

Fig. 12 gives a tracking example using a piece of the audio signal captured in San Diego, CA. Note that the ENF trace becomes weak after 15 mins, which we define as a checkpoint. AMTC can identify the trace from the noisy harmonic combined spectrum feature. Particle filter gives comparable results before the checkpoint but deviates from the true trace occasionally due to nearby interference. Local peak based tracking method YAAPT and frame-wise frequency estimator QI completely lost the target after the checkpoint as the peak information alone is not able to guarantee a good estimate.

The performance of various methods is summarized in Table IV. We calculated the mean and standard deviation of the error measures for 27 pieces of audio ENF signals. For this very noisy dataset, AMTC can achieve 0.01 Hz in offline mode and 0.03 Hz in online mode in average RMSE and 0.85 in offline mode and 0.81 in online mode in average correlation with ground truth, which outperforms all other tracking methods substantially both in average and variance of the error statistics.

VI. IMPACT OF VARIOUS FACTORS

In this section, we further evaluate the impact to the performance due to various factors. First, we study the performance when the number of spectral frames varies. Then, we discuss the effect of the trace variation level. Finally, we evaluate the impact of the distance between two traces to the estimation accuracy. The parameters are configured to be the same as introduced in Section V-A unless otherwise stated.

A. Impact of the Number of Frames

A frequency tracker starts to produce a meaningful tracking result by using two or more frames, and it is generally expected to have an improved tracking performance when more frames are used. This can be seen from the information theoretic viewpoint. Consider the true frequency state $f$ at the time instant $n$ as a random variable and denote noisy observed data at $n$th frame by $Z(n)$. Using the “conditioning reduces entropy” lemma [34] from information theory, we obtain the relationship between two posteriors $H(f(n)|Z(n), ..., Z(2), Z(1)) \leq H(f(n)|Z(n), ..., Z(2))$, where $H(\cdot)$ is the conditional entropy, suggesting less or equal uncertainty in $f(n)$ when more observations/frames are included during an inference process. Below we use experimental results to confirm that more accurate tracking results are achieved when the number of frames in the spectrogram increases.

We generated 200 trials under SNR conditions at $-16$ dB, $-14$ dB, $-12$ dB, $-10$ dB, and $-8$ dB. The duration of the test signal was set to three and a half minutes, which is equivalent to 1000 spectral frames in the spectrogram. The 1000 spectral frames were then segmented uniformly in time without overlap based on the five levels of evaluated number of frames, and the offline-AMTC was performed independently in each segment. Three performance metrics with respect to the number of frames under different SNR levels are shown in the first row of Fig. 14. We generated 200 trials under SNR conditions at $-16$ dB, $-14$ dB, $-12$ dB, $-10$ dB, and $-8$ dB. The duration of the test signal was set to three and a half minutes, which is equivalent to 1000 spectral frames in the spectrogram. The 1000 spectral frames were then segmented uniformly in time without overlap based on the five levels of evaluated number of frames, and the offline-AMTC was performed independently in each segment. Three performance metrics with respect to the number of frames under different SNR levels are shown in the first row of Fig. 14. Note that when the number of frame equals one, the tracking result using AMTC degenerates to the highest peak method. We can observe from the plots that the performance of the algorithm improves significantly when the signal length exceeds 10 frames. More frames are needed in a lower SNR condition for reaching a same performance level at a higher SNR, whereas the performance starts to converge when the number of frames equals 300 under all SNR levels.

B. Impact of Trace Variation

During the formulation process of the frequency trace tracking problem, we have assumed the change of the frequency value between two consecutive bins as a one-step discrete-time Markov chain, characterized by a transitional probability
matrix $\mathbf{P}$. With a training dataset of sufficient size available to the user, one may learn the model parameters of $\mathbf{P}$ to make a more precise tracking estimation. However, the training set is often unavailable in a real-world setting, and the user has to make their own choice of the $\mathbf{P}$ before deploying the algorithm. It is therefore important for a robust frequency tracker to successfully track the frequency components even when the variation of the frequency traces is at different levels.

We thus evaluated the system performance with respect to five different trace variation levels, and assumed the transition probability follows the uniform distribution parameterized by $k$. 200 trials were generated for each level of trace variation by tuning the variance of $f[n]$ in the generative signal model described in Section V-A1. Specifically, the five levels of the trace variation correspond to $0.001$, $0.005$, $0.01$, $0.02$, and $0.04$ bpm as the standard deviation of $f[n]$. Spectrograms of raw signals at different levels of variation before being corrupted are shown in Fig. 13. Note that a higher frequency energy diffusion is observed when the trace variation increases, as the signal within each analysis window becomes less stationary.

We show the averaged system performance in terms of $\text{RMSE}$, $\text{ERATE}$, and $\text{ECOUNT}$ with respect to different combinations of the trace variation level and the selection of $k$ in the second row of Fig. 14. The SNR was fixed to $-10$ dB. From the plots, we observe that the performance decreases when the trace variation level gets higher, especially above level III. Even though the optimal selection of $k$ increases along with the trace variation level, $\text{ERATE}$ are controlled below $5\%$ when $k$ is fixed as 4 or 6 with trace variation level lower than V, suggesting the robustness of AMTC in terms of the trace variation level with a proper selection of the transitional probability parameter.

C. Impact of Trace Distance

It is challenging for any frequency tracker to accurately distinguish and track two frequency traces that run very closely to each other especially under low SNR conditions.
To quantify the distance between two frequency components in a meaningful manner, we first defined a metric called Trace Relative Distance (TRD) as the ratio of the distance of two frequency components in the frequency domain to the mean width of their energy bumps. In Figs. 15(a)–(e), we show examples of the spectral distribution when TRD = 0.2, 0.4, 0.6, 0.8, and 1, respectively.

We generated 200 trials for each level of TRD using the same generative signal model described in Section V-A2. No “unvoiced” segment was added to the tested signal and the TRD of two frequency traces was identical over time within each tested signal. We show the averaged system performance with respect to different levels of TRD and SNR in the last row of Fig. 15. From the plots, we know that AMTC is capable to track the frequency traces with ERATE lower than 3% when SNR ≤−8 dB, and TRD ≥ 0.4. The estimation result when TRD = 0.2 is highly deviated from the ground truth. At this level of TRF, more information or prior knowledge about the frequency components is expected to be incorporated for an improved estimation.

VII. DISCUSSIONS

A. Estimation of the Number of Traces

In previous sections, we presented both the offline and the online-AMTC algorithms with the assumption that the number of traces L is known. In some cases, L is unknown and needs to be estimated. Note that the process of estimating L in the proposed AMTC system is equivalent to determining the number of iterations AMTC needs to take. The problem is then converted to deciding at which iteration should the AMTC stop. This problem can be solved by testing the hypothesis of the trace existence in the compensated spectrogram image \( Z(l) \) at each iteration \( l \).

In Section III-C, we propose to use the RER measure to detect the existence of a frequency component in each frame. We are motivated by the fact that a low RER measure of a certain frame suggests low probability of the presence of a trace in that frame. Similarly, to test globally the trace existence at \( l \)th iteration of AMTC, we propose to evaluate the average of the statistics \( \text{RER}(l) \), namely,

\[
\text{RER}(l) = \frac{1}{4} \sum_{n=0}^{3} \text{RER}(l)(n).
\]

As one example shown in Fig. 16, the ground truth number of traces in the spectrogram image is 3. We observe a significant drop in \( \text{RER}(l) \) from \( l = 3 \) to \( l = 4 \) in Fig. 16(c), when we run the offline-AMTC with four iterations. This observation coincides with the actual absence of the fourth trace. We therefore propose to estimate \( L \) as \( l - 1 \) if at the \( l \)th iteration, \( \text{RER}(l) \) is below a preset threshold. The selection of the threshold is similar to the selection of \( \Delta_{\text{RER}} \) discussed in Section V-A3.

B. Signals with Multiple Harmonics

In situations when multiple harmonic traces appear in the spectrogram (e.g., audio signals, Electrocardiography (ECG) signals), AMTC might extract several harmonic traces that originated from one single source. Take the human speech signal as an example. The fundamental frequency range of interest, 85 Hz to 255 Hz [35], [36], may cover both fundamental frequency components as well as second-order harmonics. For example, a peak in 200 Hz can be considered as the fundamental frequency component of a female speaker, or it can also represent the second-order harmonic of a male speaker. In this regard, the STFT spectrum feature might not be considered as a proper input of a robust fundamental frequency tracker. Instead, this problem can be addressed by introducing several alternative robust spectral features, e.g., the subharmonic summation method [37], the discrete logarithmic Fourier transform [38], and the frequency autocorrelation function [22]. Similar to the idea of harmonic combining algorithm [32] used for ENF case, these methods are capable of combining harmonic spectral features and improving the SNR of the fundamental frequency. The tracking performance is therefore expected to be better by feeding in any of these three features rather than the STFT spectrogram.

C. Benefits From Human-in-the-Loop Interactions

AMTC has its limitations in some specific cases. Due to the greedy nature of the searching strategy in each iteration, the algorithm may find incorrect traces when nearby strong interference is presented, or two traces with similar energies running closely in time. We show in Fig. 17(b) one such example that AMTC got confused when strong interference
is presented near the ground truth frequency trace. Note that without extra information, even humans can make mistakes in this scenario. For some applications when the analysis is performed offline, and people have prior knowledge about the trace shape or part of the trace frequency range, it is beneficial to allow users to input high-level cues [13, 39] to guide our proposed estimator’s priority to find the correct trace. As an example, Fig. [17](c) shows a user input constraint in a semi-transparent white circle for an estimated trace to pass through. Fig. [17](d) shows the constrained estimate, which was achieved by scaling up the spectrum entries in the constraint region until the estimated trace passed through the region. The constrained tracking result reveals that AMTC correctly captured the true trace by shifting its attention from interference to the user-defined region.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the problem of tracking multiple weak frequency components from spectrogram and proposed both the offline and online versions of AMTC algorithm. By iterative forward and backward dynamic trace estimation and adaptively trace carving, AMTC can provide accurate estimates even for weak frequency traces. Extensive experiments using both synthesis and real-world forensic data reveal that the proposed method outperforms several representative prior methods under low SNR conditions and can be implemented in near real-time settings. The effectiveness of the proposed algorithm may empower the development of new frequency-based forensic technologies and other small-signal applications.

REFERENCES

[1] Q. Zhu, M. Chen, C.-W. Wong, and M. Wu, “Adaptive multi-trace carving based on dynamic programming,” in 2018 Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2018, pp. 1716–1720.

[2] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142.

[3] A. Georgiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643–660, 2001.

[4] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt, “MoFa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1274–1283.

[5] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2Face: Real-time face capture and reenactment of rgb videos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2387–2395.

[6] M. C. Stamm, M. Wu, and K. R. Liu, “Information forensics: An overview of the first decade,” IEEE Access, vol. 1, pp. 167–200, May 2013.

[7] R. Garg, A. L. Varna, A. Hajj-Ahmad, and M. Wu, “‘Seeing’ ENF: Power-signature-based timestamp for digital multimedia via optical sensing and signal processing,” IEEE Transactions on Information Forensics and Security, vol. 8, no. 9, pp. 1417–1432, Sep. 2013.

[8] H. Su, A. Hajj-Ahmad, R. Garg, and M. Wu, “Exploiting rolling shutter for enf signal extraction from video,” in 2014 IEEE International Conference on Image Processing, Paris, France, Oct. 2014, pp. 5367–5371.

[9] H. Su, A. Hajj-Ahmad, C.-W. Wong, R. Garg, and M. Wu, “ENF signal induced by power grid: A new modality for video synchronization,” in Proceedings of the IEEE International Workshop on Immersive Media Experiences, Orlando, FL, Nov. 2014, pp. 13–18.

[10] W. Verkruysse, L. O. Svassand, and J. S. Nelson, “Remote plethysmographic imaging using ambient light,” Optics express, vol. 16, no. 26, pp. 21434–21445, Dec. 2008.

[11] Y. Liu, A. Jourabloo, and X. Liu, “Learning deep models for face anti-spoofing: Binary or auxiliary supervision,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, Jun. 2018, pp. 389–398.

[12] X. Li, J. Komulainen, G. Zhao, P.-C. Yuen, and M. Pietikäinen, “Generalized face anti-spoofing by detecting pulse from face videos,” in 2016 23rd International Conference on Pattern Recognition, Cancun, Mexico, Dec. 2016, pp. 4244–4249.

[13] S. Avidan and A. Shamir, “ Seam carving for content-aware image resizing,” ACM Transactions on Graphics, vol. 26, no. 3, p. 10, Jul. 2007.

[14] Q. Zhu, C.-W. Wong, C.-H. Fu, and M. Wu, “Fitness heart rate measurement using face videos,” in IEEE International Conference on Image Processing, Beijing, China, Sep. 2017, pp. 2000–2004.

[15] M. Wohlmayr, M. Stark, and F. Pernkopf, “A probabilistic interaction model for multipitch tracking with factorial hidden markov models,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 799–810, May 2011.

[16] C.-W. Wong, “Micro signal extraction and analytics,” Ph.D. dissertation, University of Maryland, College Park, 2017.

[17] W. Wang, S. Stuijk, and G. de Haan, “Exploiting spatial redundancy of image sensor for motion robust rPPG,” IEEE Transactions on Biomedical Engineering, vol. 62, no. 2, pp. 415–425, Feb. 2015.

[18] R. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Transactions on Antennas and Propagation, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[19] R. Roy and T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[20] Y. Shi and E. Chang, “Spectrogram-based formant tracking via particle filters,” in IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, Apr. 2003.

[21] R. L. Streit and R. F. Barrett, “Frequency line tracking using hidden Markov models,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 4, pp. 586–598, Apr. 1990.

[22] S. A. Zahorian and H. Hu, “A spectral/temporal method for robust fundamental frequency tracking,” The Journal of the Acoustical Society of America, vol. 123, no. 6, pp. 4559–4571, Jun. 2008.

[23] O. Ojows, J. Karlsson, J. Li, and Y. Liu, “ENF extraction from digital recordings using adaptive techniques and frequency tracking,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 4, pp. 1330–1338, May 2012.

[24] J. S. Abel, H. J. Lee, and A. P. Lowell, “An image processing approach to frequency tracking (application to sonar data),” in IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, San Francisco, CA, Mar. 1992, pp. 561–564.

[25] T. A. Lampert and S. E. O’Keefe, “An active contour algorithm for spectrogram track detection,” Pattern Recognition Letters, vol. 31, no. 10, pp. 1201–1206, Jul. 2010.

[26] Y. Liu and D. Wang, “Speaker-dependent multipitch tracking using deep neural networks,” The Journal of the Acoustical Society of America, vol. 141, no. 2, pp. 710–721, Feb. 2017.

[27] N. Esterer and P. Depalle, “A linear programming approach to the tracking of partials,” arXiv preprint arXiv:1901.05044, 2019.

[28] J. Neri and P. Depalle, “Fast partial tracking of audio with real-time capability through linear programming,” in 21st International Conference on Digital Audio Effects, 2018.
[29] D. Rife and R. Boorstyn, “Single tone parameter estimation from discrete-time observations,” *IEEE Transactions on Information Theory*, vol. 20, no. 5, pp. 591–598, Sep. 1974.

[30] M. Wu, D. Wang, and G. J. Brown, “A multipitch tracking algorithm for noisy speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, May 2003.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[32] A. Hajj-Ahmad, R. Garg, and M. Wu, “Spectrum combining for ENF signal estimation,” *IEEE Signal Processing Letters*, vol. 20, no. 9, pp. 885–888, Sep. 2013.

[33] J. O. Smith and X. Serra, PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. CCRMA, Department of Music, Stanford University, 1987.

[34] T. M. Cover and J. A. Thomas, “Elements of information theory.” John Wiley & Sons, 2012, ch. 2.

[35] I. R. Titze, *Principles of Voice Production*. Englewood Cliffs, N.J., Prentice-Hall, 1994.

[36] R. J. Baken and R. F. Orlikoff, *Clinical Measurement of Speech and Voice*. Cengage Learning, 2000.

[37] D. J. Hermes, “Measurement of pitch by subharmonic summation,” *The Journal of the Acoustical Society of America*, vol. 83, no. 1, pp. 257–264, 1988.

[38] C. Wang and S. Seneff, “Robust pitch tracking for prosodic modeling in telephone speech,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Istanbul, Turkey, Jun. 2000, pp. 1343–1346.

[39] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra, “Halftone QR codes,” *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 217: 1–8, Nov. 2013.