Fusion of Live Audio Recordings for Blind Noise Reduction

Aaron Ballew
ballew@u.northwestern.edu

Aleksandar Kuzmanovic
akuzma@northwestern.edu

Chung Chieh Lee
cclee@northwestern.edu

***Abstract*—The prevalence of digital cameras and video-capable mobile phones enables the common practice of audiences capturing recordings of live music performances. It is now increasingly common to find some of these personal recordings online, uploaded to popular video hosting websites. Recognizing the desire of music fans to obtain a recording of good audio quality, we offer a time-domain fusion technique for combining these samples to achieve higher signal-to-noise ratio (SNR) than the single best original sample. When no composite output can improve SNR, the best original sample recording is returned.**

The scenario is modeled as a single blind source with multiple diverse receivers. As every live performance is unique, we assume no prior knowledge of a reference signal, and no knowledge of the original recordings' SNRs. Using statistical characteristics among the samples, we infer relative SNR, rank samples by quality, and determine whether a composite delivers improvement. The technique can be applied in a variety of contexts where multiple receivers have opportunity to capture audio, speech, or other signals.

Keywords: Audio, acoustic noise, blind signal, estimation.

I. Introduction

The concert-going experience now commonly includes the sight of hundreds of digital cameras and mobile phones aimed at the stage. Audience members record their experience for later enjoyment, and increasingly share them online. On popular video hosting websites like YouTube, it is not unusual to find five to ten recordings of a popular song within days of its performance. These recordings represent multiple vantage points on the same song performed at the same event.

Recognizing that music fans get unique enjoyment from hearing recordings of a performance they personally attended, we attempt to generate an improved audio recording from the raw audience-generated content. Given a sample set, i.e., a collection of available recordings of the same event, our goal is to identify the "best" raw sample, and generate a noise-reduced composite that is "better" than the original "best."

The problem is modeled as a single blind source with multiple diverse receivers and unknown SNR. Consider the following scenario. Signal is generated from stage-speakers. Typically the signal is broadcast as a mono mix, and even if not, receivers record multi-tracking as mono. The speakers may be spread apart on-stage, but relative to the venue's scale they are quite close together resulting in synchronization from a listener's perspective (though various recordings need not be synchronized to each other). Receivers are distributed arbitrarily throughout the audience, all capturing a noise-corrupted representation of exactly the same underlying music signal. Noise occurs in the form of cheering, clapping, shouting, in addition to the expected static, multi-path fading, etc. Noise near a particular receiver is not the same noise near another receiver, unless both receivers happen to be close together. This local noise will not reach distant receivers, because it is drowned out by noise local to the distant receivers. (Note, this assumption is inappropriate in the context of a quiet auditorium, where one cough may appear on every recording.) Finally, because every live performance is unique (e.g., different tempo, arrangements, embellishments, etc.), there is no prior reference of what part of each recording comes from the signal and what part comes from noise, thus the SNR is unknown. Even a noiseless studio recording does not provide adequate reference, since live performances often deviate significantly. The music may even purposely incorporate some amount of clapping or shouting that serves to further blur the distinction between signal and noise.

This problem is not easily addressed by established techniques. Methods of estimating unknown SNR, such as those found in digitial signal processing, typically assume prior knowledge of M-ary waveforms [9]. In the current scenario, there is neither an absolute reference waveform nor any opportunity to pre-code the signal's waveform prior to broadcast. Independent component analysis (ICA) is often applied to blind separation of independent audio sources, e.g., overlapping speech signals or musical instruments [8]. In the current scenario, our aim is to selectively enhance a single source, itself composed of many instruments, among uncountably many noise sources. This is not a conceptual fit for the separation problem, nor are there enough receivers to satisfy the requirements of ICA's mathematical model.

Instead, our problem belongs to the broad area of sensor fusion, in which data captured from disparate sources is combined to achieve better outcomes than the individual sources offer alone. In a general sense, fusion is applied to improve decisions [4] [11]. This includes detection, localization, and has also proven a useful tool in the enhancement of images [12]. Speech enhancement has also benefitted from fusion, for example in [1], where phase deviations between two microphones are used to reduce noise. [7] builds on this technique and points out some challenges of not having a-priori knowledge of input SNR. Returning to the current scenario, we offer a complement to existing lines of work while tackling the large problem of enhancing a source composed of many individual voices and instruments.

In this paper we propose a time-domain fusion technique for combining raw sample audio recordings in a way that preserves the *sameness* among the samples while reducing the *differentness*. The scope is not limited to speech, and is specifically presented in the context of complex music signals. Improvement depends on the uncorrelatedness of the noise. Correlation among the noise results in confusion with the signal, so that portion of noise cannot be reduced. However, listeners already accept some amount of crowd noise in live recordings, and even a modest improvement is of value to fans. Translation of time-domain fusion to a corresponding frequency-domain approach offers comparable results.

We believe that making this tool available as an application will encourage more sharing of live recordings, leading to larger sample sets with the potential for more significant noise reduction. However, the technique's applicability is not limited to entertainment. Any context in which unknown audio, speech, or other signals are captured by multiple receivers is a potential application. Such contexts can range from wireless communications to espionage.

In Section II we introduce characteristics of sample sets allowing for an improved composite. Three idealized cases enable analysis of potential SNR gain. One case is then selected as the basis of a signal fusion algorithm. In Section III, the algorithm synchronizes, normalizes, ranks, and optimally combines samples. The resulting output is either the improved composite, or, if the composite offers no improvement, the best single sample from the original set. Of special importance is a novel iterative normalization procedure that successfully equalizes signal powers (as opposed to total powers) among samples without prior knowledge of SNR. In Section IV, we investigate real noise and discuss the fusion algorithm's effect on simulated and real samples.

II. SNR Gain Analysis

Here we establish conditions under which audio samples can be averaged to enhance SNR. This is related to work done in the context of averaging noisy images [12]. Three idealized cases offer tractable SNR analysis. Since real sample sets are not expected to fit neatly into any of the cases [2], the signal fusion algorithm of Section III transforms the samples into conformance with Case II, Section II-B, provided that sample noise is uncorrelated.

Assume sample recordings are already of equal length, perfectly synchronized, and ranked in order of decreasing SNR. Section III describes how this is achieved in real samples. Other details impacting real-world implementation, such as encoding, sampling rate, etc., are also discussed in Section III, under Data Preparation. Assume that noise is white in each sample and uncorrelated between samples. Section IV investigates the characteristics of real noise.

Let N be the number of available samples, x_i, where $i = 1, 2 \ldots N$. Let n_i be uncorrelated white noise corrupting an instance of the transmitted signal s_i, such that $\text{Cov}(n_i, n_j) = 0$, $i \neq j$ and $\text{Cov}(s_i, n_i) = 0$, $\forall i$. Then $x_i = s_i + n_i$ and $\sigma^2_{x_i} = \sigma^2_{s_i} + \sigma^2_{n_i}$.

A. Case I: Identical signals, equal noise powers

This is an idealized scenario in which each sample x_i contains an identical signal with power σ^2_s and independent noise n_i with equal power σ^2_n. Direct averaging returns the composite $\overline{x}$ with unchanged signal power but reduced noise power proportional to the number of samples N.

$$\text{Assume} \begin{cases} s_i = s,\ \text{Cov}(n_i, n_j) = 0 \text{ for } i \neq j \\ \sigma^2_{s_i} = \sigma^2_s,\ \sigma^2_{n_i} = \sigma^2_n \end{cases}$$

Then, the signal power and noise power of the sample average are given by

$$\sigma^2_{\overline{s}} = \frac{1}{N^2}\text{Var}(Ns) = \sigma^2_s \tag{1}$$

$$\sigma^2_{\overline{n}} = \frac{1}{N^2}\sum_{i=1}^{N}\text{Var}(n_i) = \frac{\sigma^2_n}{N} \tag{2}$$

Thus, an SNR gain of N is attained due to multireceiver fusion by direct averaging of N independent identically distributed samples [3].

B. Case II: Identical signals, unequal noise powers

Now relax the noise constraint of Case I by allowing each sample x_i's noise power $\sigma^2_{n_i}$ to differ from σ^2_n by some scaling factor k_i. Assume $k_1 = 1$, and $k_1 \leq k_2 \leq \ldots \leq k_N$. This sets the least noisy sample as a benchmark for comparison. All samples x_i still contain the same signal power σ^2_s. Direct averaging returns $\overline{x}$ with unchanged signal power but a noise power that is increased or decreased depending on the growth rate of k_i. It can be shown that if k_i grows too fast, noise contributes too much to the composite and overwhelms the effect of Section II-A. If k_i grows slowly enough, then improved SNR is possible from averaging. Finally, the largest SNR gain may result from combining only a subset of the samples.

By having equal signal powers among all samples, the differences among the samples' total powers $\sigma^2_{x_i}$ are due exclusively to the noise components $\sigma^2_{n_i}$. The ability to use total power as a proxy for noise power is a significant convenience when the actual signal power is unknown. It allows us to rank samples in order of quality and judge whether a composite results in more or less noise power than any of the original samples. The critical task then becomes normalizing real samples such that all have equal signal power, even if it is not known exactly what that signal power is. Normalization with unknown SNR is achieved through an iterative procedure described in Section III-D.

Let M be the number of samples actually incorporated into the composite such that the resulting noise power is minimized, where $1 \leq M \leq N$.

$$\text{Assume} \begin{cases} s_i = s,\ \text{Cov}(n_i, n_j) = 0 \text{ for } i \neq j \\ \sigma^2_{s_i} = \sigma^2_s,\ \sigma^2_{n_i} = k_i \cdot \sigma^2_n \end{cases}$$

Then, the signal power and noise power of the sample average are given by

$$\sigma_{\bar{s}}^2 = \frac{1}{M^2}\text{Var}(Ms) = \sigma_s^2 \tag{3}$$

$$\sigma_{\bar{n}}^2 = \text{Var}\Big[\frac{1}{M}(\sqrt{k_1}n_1 + \sqrt{k_2}n_2 + \ldots + \sqrt{k_M}n_M)\Big] \tag{4}$$

$$= \frac{1}{M^2}\Big[k_1\sigma_n^2 + k_2\sigma_n^2 + \ldots + k_M\sigma_n^2\Big] \tag{5}$$

$$= \frac{\sigma_n^2}{M^2}\sum_{i=1}^{M} k_i \tag{6}$$

Thus, whereas the original best sample x_1's SNR is σ_s^2/σ_n^2, the SNR of the composite is $(M^2/\sum_{i=1}^{M} k_i) \cdot (\sigma_s^2/\sigma_n^2)$. SNR is improved over the original best sample x_1 when $(M^2/\sum_{i=1}^{M} k_i) > 1$. For example, a growth rate of $k_i < 2i-1$ is a sufficient condition to meet the threshold for improvement.

C. Case III: Unequal signal powers, equal noise powers

This third idealized scenario introduces assumptions which at first glance are a slight alteration of Case II, but result in unwieldy complications. Assume that the noise components n_i of each sample x_i have equal power, i.e., $\sigma_{n_i}^2 = \sigma_n^2$. Assume each sample contains the same signal but of different powers, i.e., $\sigma_{s_i}^2 = k_i \cdot \sigma_s^2$. Without loss of generality, assume $k_1 = 1$, and $k_1 \geq k_2 \geq \ldots \geq k_N$. Averaging now changes both the signal and noise powers simultaneously.

Let M be the number of samples actually incorporated into the composite such that the resulting SNR is maximized, where $1 \leq M \leq N$.

$$\text{Assume} \begin{cases} s_i = \sqrt{k_i} \cdot s,\ \text{Cov}(n_i, n_j) = 0 \text{ for } i \neq j \\ \sigma_{s_i}^2 = k_i \cdot \sigma_s^2,\ \sigma_{n_i}^2 = \sigma_n^2 \end{cases}$$

Then, the signal power of the sample average is given by

$$\sigma_{\bar{s}}^2 = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\text{Cov}(\sqrt{k_i}s, \sqrt{k_j}s) \tag{7}$$

$$= \frac{\sigma_s^2}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\sqrt{k_i k_j} \tag{8}$$

From Eq.2, the composite noise power becomes $\sigma_{\bar{n}}^2 = \sigma_n^2/M$. Whereas the best sample's SNR is σ_s^2/σ_n^2, the composite's SNR becomes $\big(\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{M}\sqrt{k_i k_j}\big) \cdot (\sigma_s^2/\sigma_n^2)$.

Although this case appears similar to Case II, fitting real samples with unknown SNR into Case III proves difficult to work with. Mainly, we must normalize noise power in the samples, which is impossible. Further, we must track the composite's changing signal power while its noise power changes simultaneously. For these reasons, Case II is selected for algorithmic implementation as described in Section III.

III. Processing and Fusing Algorithm

The algorithm that deals with real music samples consists of five main stages: data preparation, selecting a reference and synchronization, windowing, normalization and ranking, and combining. These steps transform the samples to fit the assumptions of Case II, Section II-B. Explanations are provided in a mix of math notation and MATLAB-inspired pseudocode. Noise is assumed uncorrelated between samples.

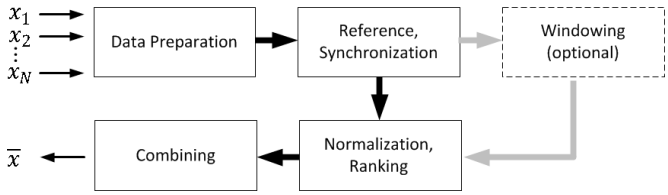


Figure 1. Block diagram of algorithm stages.

To acquire real sample recordings, we have used our own custom-written search application that interfaces with YouTube via API. This application takes standard search terms as its input, such as song title, city, venue, date, or artist. It delivers the search query to YouTube, resulting in a comprehensive list of potential matches. Then, applying TF-IDF [10] and cluster analysis [5] to the files' meta data, it sorts the results into groups likely to be recordings of the same performance. For example, a search for a particular song title and the city "New York" may return performances of that song spanning the artist's many different visits to New York. The application distinguishes the performance of June 1, 2010 from the performance of March 1, 2011. Detailed explanation of these mechanisms are beyond the scope of this paper. After successful grouping, the files of interest are downloaded on-demand (e.g., five recordings of the song performed in New York on March 1, 2011).

A. Data Preparation

We begin with a set of N raw samples with unknown SNRs. Each sample x_i is a time series stored as a single column (mono) vector of amplitudes. Samples are not necessarily of equal vector lengths. In practice this is the effect of converting YouTube MP4 files to single track audio WAV format. One advantage of using YouTube as the primary source of content is that live mono audio recordings are typically stored with one of only a few sampling rates (22.05kHz or 44.1kHz), and this parameter can be read from meta data. We identify the maximum length vector, and zero-pad the remaining vectors such that all vectors are of equal length. Let this equal length be denoted L, for later use.

B. Reference Selection And Synchronization

Lacking an external reference of what a "good" sample is, we use an internal reference chosen from the sample set. This is accomplished by computing the correlation matrix P of the sample set and summing across rows (or columns). The row with the largest sum represents the sample with the most

similarity to all other samples, i.e., the highest SNR, since noise components do not correlate to each other. This sample is designated as the reference. In practice, we have observed that this procedure consistently selects a good quality sample.

Direct computation of P is not possible while the samples are unsynchronized. So, we construct each entry ρ_{ij} of the correlation matrix from the maximal cross-correlation among each sample pair. This returns both the synchronized correlation ρ_{ij} and the corresponding lag τ_{ij} that would achieve synchronization.

for $i = 1 : N$ **do**
  for $j = 1 : N$ **do**
    $[\rho_{ij}, \tau_{ij}]$ = max(xcorr(x_i, x_j))
  end
end

Then, sum along rows (or columns) of P,

for $i = 1 : N$ **do**
  $\text{row}_i = \sum_{j=1}^{N} \rho_{ij}$
end

where i corresponding to max(row) identifies the reference sample. We call this sample x_{ref}, and perform a circular vector shift of each remaining sample according to the already computed $\tau_{\text{ref},j}$. Now all samples are synchronized to the reference.

C. Windowing

In practice, a sample's noise power may vary with time. For example, an audience member may move his cellphone such that a low-noise recording suddenly changes to high-noise, or vice versa. This is a form of low-frequency noise that contributes to an unstable sample being treated as more noisy overall. To help avoid penalizing the low-noise segment of such a sample, the algorithm may optionally process synchronized sample sets as a sequence of shorter-duration windows. Each window effectively becomes a new instance of the fusion problem. This stabilizes the noise power within each window's duration while creating freedom for adjacent windows to return a composite output uniquely "best" for its time period within the overall recording. The composite outputs of each window are concatenated prior to playback.

Recall that all N synchronized samples are of equal length L, as a result of Section III-A. With the windowing option enabled, the samples are segmented into shorter-duration windows of length L/W_t data points, where W_t indicates windowing in the time domain. This results in W_t adjacent windows, each containing N synchronized excerpts, or subsamples, of the original full recordings. Each window then becomes its own instance of the fusion problem (Fig. 2). Subsequent processing steps apply unchanged, without loss of generality, to each individual window of length L/W_t, for $W_t = 1 \ldots L$. Adjacent windows may then return different composite outputs based on the noise conditions specific to that time period.

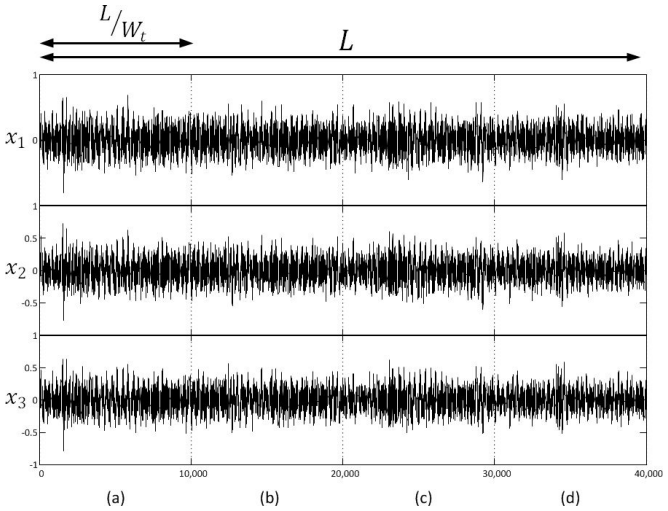


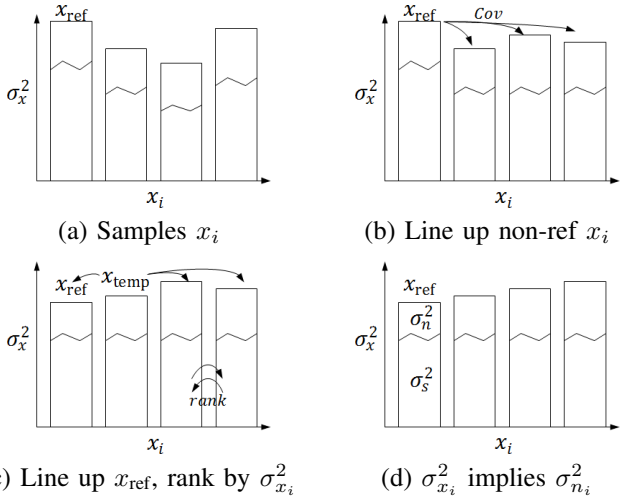
Figure 2. $N = 3$ synchronized time-series samples x_i of total length $L = 40,000$ data points, segmented into $W_t = 4$ windows (a), (b), (c), (d) of length $L/W_t = 10,000$.

Note that there is a trade-off as windows become exceedingly short in duration, in that uncorrelated noise tends to become increasingly correlated with fewer data points. This implies there should be an optimal window size that balances noise correlation against time-dependency. Recommending the optimal window size is the subject of ongoing work, and for now window size is left as a tunable parameter.

Figure 2 depicts an approximately 1s excerpt (40,000 data points sampled at 44.1kHz) of three synchronized audio samples. In this example, segmenting each time-series into windows of $10,000$ data points results in four instances of the fusion problem. Each instance operates on three synchronized audio samples with a duration of approximately $\frac{1}{4}$s.

D. Normalization and Ranking

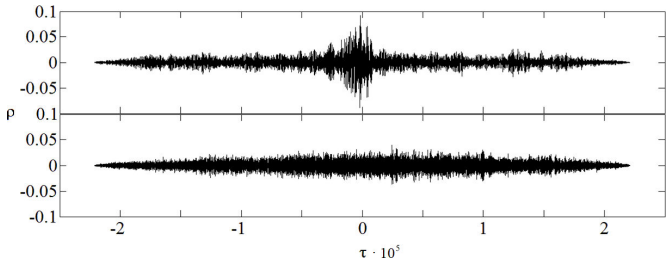
All windowed samples are now of equal length L/W_t data points ($W_t = 1$ if windowing is not enabled) and time-synchronized to a reference, but with unequal total powers and unknown SNRs. Absolute SNR cannot be known, but total power can be used to infer relative noise power among the samples under certain conditions. Specifically, this is possible only when all samples' signal powers are equal, as defined in Section II-B, Case II. Thus it is necessary to scale samples and measure whether the signal components are of equal power. Correlation as applied in Section III-B is insensitive to scale, i.e., multiplying a sample by a scaling factor does not change ρ_{ij}. Instead, we use covariance as the measurement. If noise is uncorrelated, then covariance between two samples is proportional to the signal components' strengths. This leads to a novel iterative approach to normalizing signal powers and consequently measuring relative SNR among samples when absolute SNR is unknown.



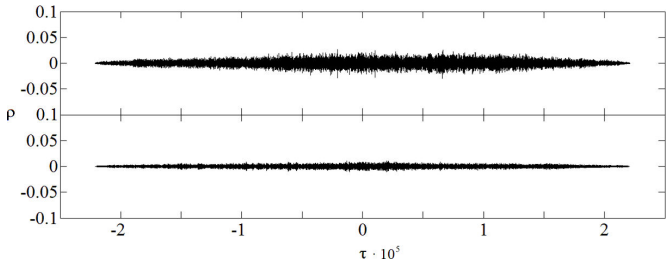
(a) Samples x_i (b) Line up non-ref x_i

(c) Line up x_{ref}, rank by $\sigma_{x_i}^2$ (d) $\sigma_{x_i}^2$ implies $\sigma_{n_i}^2$

Figure 3. Normalization & ranking of unknown signal powers.



(a) Sample$_1$ vs. sample$_2$ (top), noise$_1$ vs. noise$_2$ (bottom)



(b) Sample$_1$ vs. noise$_2$ (top), sample$_2$ vs. noise$_1$ (bottom)

Figure 4. Cross-correlation ρ vs. lag τ.

Given $\text{Cov}(x_{\text{ref}}, x_i) = c_i$, and $\text{Cov}(x_{\text{ref}}, x_j) = c_j$,

$$\text{Cov}(x_{\text{ref}}, x_i) = c_i \equiv h_j \cdot c_j \quad (9)$$
$$= h_j \cdot \text{Cov}(x_{\text{ref}}, x_j) \quad (10)$$
$$= \text{Cov}(x_{\text{ref}}, h_j x_j) \quad (11)$$

Scaling x_j by h_j (the ratio of the covariances) results in x_i and $h_j x_j$ having equal covariance to the reference, x_{ref}. This relationship can be used to scale all non-reference samples such that they have equal covariance to x_{ref}. Without loss of generality, assume $x_{\text{ref}} = x_1$ and $\text{Cov}(x_{\text{ref}}, x_2) = c_2$. Finding h_j such that $\text{Cov}(x_{\text{ref}}, h_j x_j) = c_2$ for $2 \leq j \leq N$ results in all scaled non-reference samples having the same signal power (Fig.3b). Finally, $x_{\text{ref}} = x_1$ must be scaled such that its signal power equals that of the non-reference samples. Now note $\text{Cov}(h_i x_i, h_j x_j) = \text{constant} \equiv C$, for $i, j = 2 \ldots N, i \neq j$. We temporarily select the scaled non-reference sample with lowest total power as a new reference, x_{temp}, and find h_1 such that $\text{Cov}(x_{\text{temp}}, h_1 x_1) = C$.

Now all samples have equal signal power, though the exact value of the signal power remains unknown (Fig.3c). This process is loosely inspired by [6].

With signal power normalized, it is straightforward to rank the samples in order of increasing total power, and thus increasing noise power, completing the prerequisites of Case II (Fig.3d).

E. Combining

With all samples normalized and ranked, we compute the average of the first M samples such that total power $\sigma_{\overline{x}}^2$ is minimized. This is the new "best" composite, $\overline{x}$. Ranking eliminates the need to evaluate many combinatorial subsets, as there are no more than $M = N$ averages to consider, and $M = 1$ is simply x_{ref} itself. If $M = 1$, then no composite was successful in surpassing the SNR of the original "best" sample, x_{ref}.

for $m = 1 : N$ **do**
  power-of-avg$_m$ = Var$(\frac{1}{m} \sum_{i=1}^{m} h_i x_i)$
end
$[\sigma_{\overline{x}}^2, M]$ = min(power-of-avg)

Recall that the composite only offers improvement when the threshold $(M^2 / \sum_{i=1}^{M} k_i) > 1$ is met, as described in Section II-B. An attractive outcome of the algorithm is that even when the threshold is not met, the original "best" sample is identified. This feature alone is of value to users who would otherwise manually search through many poor quality recordings before finding one of good quality.

IV. Characterization of Noise Reduction

These processes depend on noise being uncorrelated among samples. Inspection of real samples suggests this assumption is imperfect but acceptable. Figure 4 shows an example of cross-correlations between two typical song samples. Sample-to-sample correlation is strong, whereas noise-to-sample and noise-to-noise correlation is weak. Noise-only clips come from the song samples just before or after music is performed.

We have applied the algorithm to samples with real noise and to studio recordings with artificial noise. In both cases, synchronization and normalization are effective. Artificial noise is noticeably reduced. On real samples, perceived noise reduction varies depending on the sample set and listener. Interested readers may visit networks.cs.northwestern.edu/∼aaron/fusion to hear before-and-after examples.

To visually illustrate blind noise reduction, Figure 5 (top) shows six waveforms ranked and concatenated left to right in order of decreasing noise power. A noiseless studio recording was corrupted with varying levels of noise to generate four

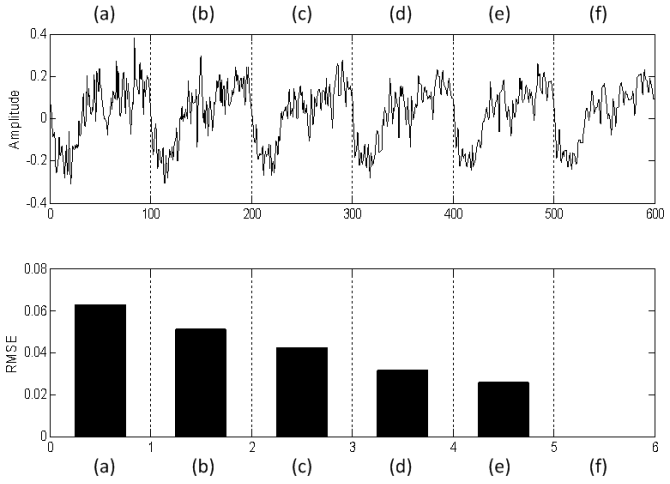


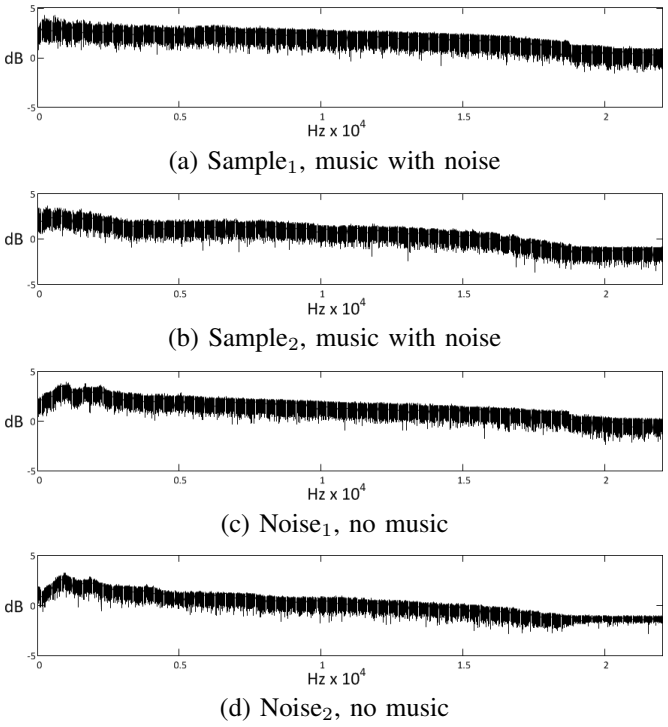
Figure 5. Identical time-series interval of 100 data points repeated six times with differing amounts of noise (top), and corresponding RMSE (bottom). (a–d) Left to right, clips ranked in order of decreasing noise, (e) noise-reduced composite, and (f) noiseless studio recording

noisy samples. These noisy samples were fed through the fusion algorithm, resulting in a noise-reduced composite. Each window of Figure 5 represents 100 data points excerpted from the same respective time interval of each sample recording. Intervals (a) through (d) are the ranked noise-corrupted input samples. Interval (e) is the noise-reduced composite returned by the signal fusion algorithm. Interval (f) is the original noiseless studio recording. Figure 5 (bottom) indicates each interval's root mean squared error (RMSE) relative to the noiseless studio recording. The fused composite of interval (e) offers the least RMSE relative to the noiseless reference.

V. Extension into the Frequency Domain

Though the technique presented in this paper operates in the time-domain, it is also instructive to consider the noise spectrum. Noise uncorrelated in the time domain remains uncorrelated in the frequency domain. This noise need not be strictly white, and its energy may appear stronger in certain frequency ranges. So long as the threshold $(M^2/\sum_{i=1}^{M} k_i) > 1$ is satisfied in a given frequency range, noise reduction is possible in that frequency range regardless of the spectrum's overall shape. Figure 6 shows frequency responses of selected real music and noise samples. The sampling rate represented is 44.1kHz, thus frequency content ceases beyond 22.05kHz. Noise occupies the entire bandwidth of the music-plus-noise samples. Noise energy is fairly flat but exhibits a consistently "pink" character [13], having a stronger low frequency component that smoothly rolls off toward the higher frequencies. This is a characteristic typical of speech and audio that persists across samples.

Now consider a case where noise does not exhibit a consistent overall spectral shape among the samples (e.g., not all white or all pink). For example, assume a particular sample has almost no low frequency noise, but considerable high frequency noise, while all other samples' noise is pink. This sample's high quality audio content in the lower frequency



(a) Sample$_1$, music with noise

(b) Sample$_2$, music with noise

(c) Noise$_1$, no music

(d) Noise$_2$, no music

Figure 6. Frequency spectrum of real music and noise samples, 0–22.05 kHz. Phase plots not shown.

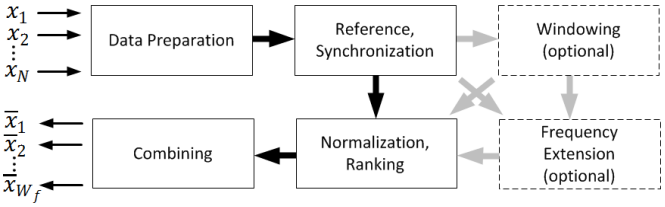
range will not benefit from fusion with other noisy samples. However, its noisy high frequency content could benefit from fusion with the high frequency noise of the other samples.

To accommodate this, we adapt the time-domain fusion algorithm to operate separately on frequency sub-bands. This is analogous to Section III-C's time-domain windowing. Note that a performance advantage over the pure time-domain approach is not expected unless noise exhibits inconsistent spectra among samples.

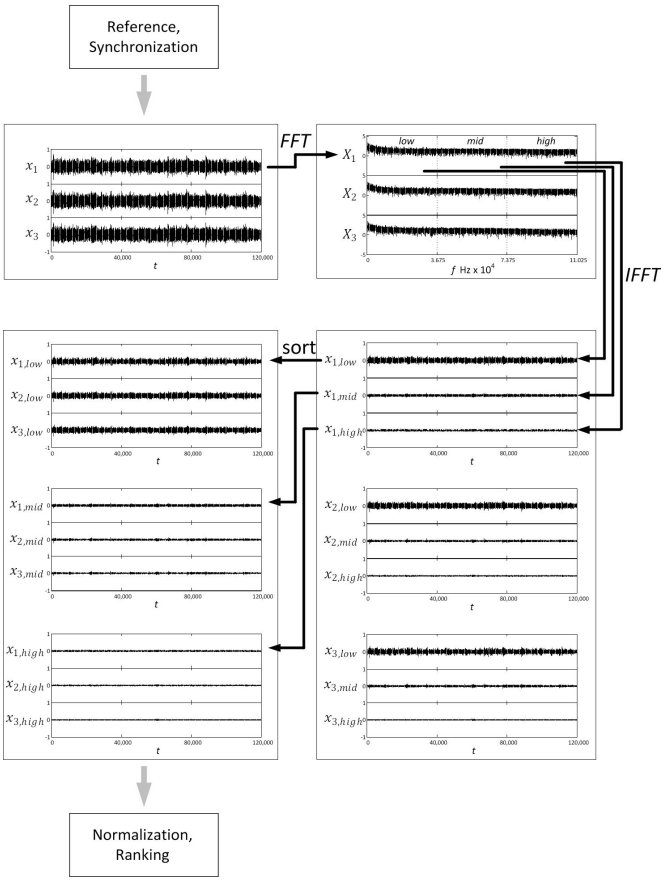
Let L be the length of each synchronized input sample vector x_i, and let W_f be the number of windows in the frequency domain. Let $\mathscr{F}$ and $\mathscr{F}^{-1}$ indicate the FFT and IFFT operators, respectively.

for $i = 1 : N$ **do**
  $X_i = \mathscr{F}\{x_i\}$
  for $j = 1 : W_f$ **do**
    $X_{ij} = X_i[(j-1)\frac{L}{W_f} + 1 : j\frac{L}{W_f}]$
    $x_{ij} = \mathscr{F}^{-1}\{X_{ij}\}$
  end
end

Figure 7a illustrates where this new stage fits in the overall block diagram. After synchronization, all samples are processed through FFT, windowed into W_f frequency sub-bands of width L/W_f, and then converted back to the time domain by IFFT. Each original sample x_i has now been decomposed into W_f time domain sub-samples x_{ij}, with j indicating which frequency sub-band is represented. For each j, the set of



(a) Block diagram with frequency-domain extension



(b) Frequency-domain stage with x_1 flow highlighted

Figure 7. Each incoming x_i undergoes FFT, followed by segmentation into $W_f = 3$ frequency ranges. In this example, *low*, *mid*, and *high* ranges are defined (phase not shown). Each frequency segment undergoes IFFT, followed by sorting into groups corresponding to like ranges. The three groups, $x_{i,low}$, $x_{i,mid}$, and $x_{i,high}$ are treated as distinct fusion instances fed separately to the normalization stage.

sub-samples x_{ij}, $i = 1 \ldots N$ constitutes a distinct instance to be processed by the fusion algorithm. The outputs of the combining stage become $\overline{x}_j$, the new "best" representations *of each particular frequency sub-band*. Figure 7b illustrates the decomposition of three input time series into nine time series grouped according to low, medium, and high frequency. These sub-samples $\overline{x}_j$, still synchronized, are ultimately recombined by weighted addition in the time-domain, where any reasonable weighting scheme may be applied (not illustrated). For example, each sub-sample $\overline{x}_j$ may be scaled such that the overall spectrum of the final recombined output $\overline{x}$ adheres to a white or pink shape.

Returning to the example of a sample with little low frequency noise, its good quality audio from the low frequency range may be returned unaltered by the algorithm, while the noisy high frequency component may be fused with other samples for noise reduction. The preceding discussion assumes equal-width windows, but this is not required. Variable windows, e.g., by octave corresponding to a pink spectrum, are possible but beyond the scope of this paper. Importantly, the frequency-domain extension gives the algorithm the ability to address a wider variety of noise characteristics while still operating fundamentally in the time-domain.

VI. Conclusion

This paper takes advantage of real-world user behavior and statistical signal processing to achieve signal enhancement without conventional inputs. No reference signal, waveform, or SNR is known beforehand. Samples are successfully synchronized, normalized by signal power, and combined such that the composite's SNR is maximized. If no composite achieves improvement, the best original sample is returned.

Audio quality enhancement is of clear benefit to users. Equally important is the automation of an otherwise manual search for the "best" single sample among many. Users save time while getting the assurance of a listenable recording. In addition, the ability to equalize signal powers among multiple receivers without prior knowledge of SNR may be useful in many other areas of statistical signal processing. It is hoped that the availability of this service will encourage more user-generated sharing of sample recordings, leading to further improvements in audio quality.

References

[1] P. Aarabi and S. Guangji, "Multi-Channel Time-Frequency Data Fusion," in *Information Fusion*, vol. 1, 2002, pp.404–411.

[2] B. Barrow, "Diversity Combination of Fading Signals with Unequal Mean Strengths," in *IEEE Communications Systems*, vol. 11, no. 1, March 1963, pp.73–78.

[3] D.G. Brennan, "Linear Diversity Combining Techniques," in *Proc. of the IEEE*, vol. 91, no. 2, pp. 331–356, Feb 2003.

[4] B. Dasarathy, *Decision Fusion*, IEEE Computer Society, USA, 1994.

[5] R. Gil-Garcia, J.M. Badia-Contelles, and A. Pons-Porrata, "A General Framework for Agglomerative Hierarchical Clustering Algorithms," in *Pattern Recognition*, 2006, pp.569–572.

[6] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," in *SIGCOMM Comp. Comm. Rev.*, vol. 38, no. 4, 2008, pp.159–170.

[7] S. Guangji, P. Aarabi and N. Lazic, "Adaptive time-frequency data fusion for speech enhancement," in *Information Fusion*, 2003, pp.394–399.

[8] T-W. Lee, A.J. Bell and R. Orglmeister, "Blind Source Separation of Real World Signals," in *Proceedings of IEEE International Conference on Neural Networks*, Houston, June 1997, pp.2129–2135.

[9] D. Pauluzzi and N. Beaulieu, "A Comparison of SNR Estimation Techniques for the AWGN Channel," in *IEEE Transactions on Communications*, vol. 48, no. 10, October 2000, pp.1681–1691.

[10] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval,"in *Information Processing and Management*, vol. 24, no. 5, 1988, pp.513–523.

[11] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[12] M. Unser and M. Eden, "Weighted averaging of a set of noisy images for maximum signal-to-noise ratio," in *ICASSP*, vol. 38, no. 5, May 1990, pp.890–895.

[13] R. Voss and J. Clarke, "1/f Noise in Music and Speech," in *Nature*, vol. 258, no. 5533, November 1975, pp.317–318.