COMB-FILTER FREE AUDIO MIXING USING STFT MAGNITUDE SPECTRA AND PHASE ESTIMATION

Volker Gnann and Martin Spiertz

Institut für Nachrichtentechnik
RWTH Aachen University
Aachen, Germany
{gnann,spiertz}@ient.rwth-aachen.de

ABSTRACT

This paper presents a new audio mixing algorithm which avoids comb-filter distortions when mixing an input signal with time-delayed versions of itself. Instead of a simple signal addition in the time domain, the proposed method calculates the short-time Fourier magnitude spectra of the input signals and adds them. The sum determines the output magnitude on the time-frequency plane, whereas a modified RTISI algorithm estimates the missing phase information. An evaluation using PEAQ shows that the proposed method yields much better results than temporal mixing for non-zero delays up to 10 ms.

1. INTRODUCTION

The purpose of audio mixing is to take a given number $C \in \mathbb{N}$ of input signals $x_1(n), \cdots, x_C(n)$, to assign a weight $a_c \in \mathbb{R}_0^+$ to each input signal $x_c(n)$, and to calculate an output signal which merges the input signals. We can easily extend this concept to multiple output channels. The traditional approach is to calculate the output signal $x(n)$ as a linear combination of the input signals:

$$x(n) = \sum_{c=1}^{C} a_c x_c(n). \tag{1}$$

In the following, we call this approach "temporal mix" because it is calculated in the time domain. The temporal mix leads to problems when we record a single audio source using multiple microphones on different positions. Due to different distances between the sound source and each microphone, respectively, the sound waves need less time to propagate to the first microphone than to the second one (see Figure 1). When we add ("mix") the signals of both microphones, the impulse response and the transfer function of the resulting system are

$$h(t) = a_1\delta(t) + a_2\delta(t-\Delta t), \tag{2}$$

$$H(f) = a_1 + a_2 e^{-j2\pi f \Delta t}. \tag{3}$$

In the case of $a_1 = a_2 = 1$, the magnitude frequency response becomes:

$$|H(f)| = \sqrt{2 + 2\cos(2\pi f \Delta t)} \tag{4}$$

Figure 2 illustrates this response in dB. The response is characterized by +6dB "peaks" on the positions

$$f_k^{\text{peak}} = \frac{k}{\Delta t},\ k \in \mathbb{N}_0, \tag{5}$$

and by "notches" (interference cancellations) at the positions

$$f_k^{\text{notch}} = \frac{k+0.5}{\Delta t},\ k \in \mathbb{N}_0. \tag{6}$$

Due to this frequency response, the resulting effect is called "comb filter".

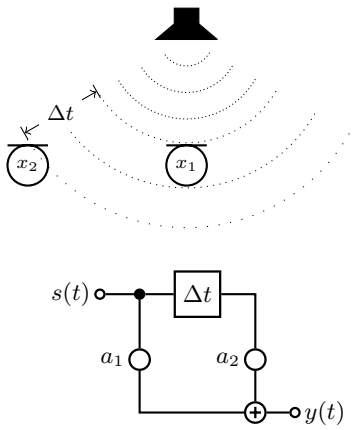

Figure 1: *Example of an acoustic comb filter and its equivalent system.*

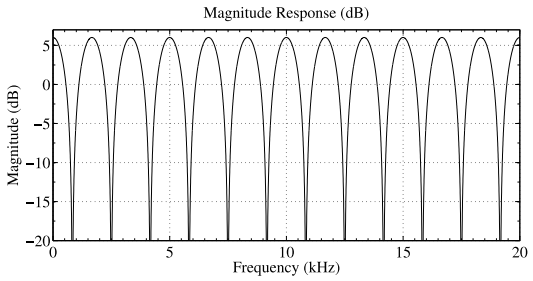

Figure 2: *Comb filter frequency response. Δt=0.5 ms, a_1=a_2=1.*

Comb-filter distortions can lead to sound discolorations and thus should be avoided. Brunner and others [1] carried out listening tests with the result, that — on average and under good listening conditions — comb filter distortions with level differences of 18 dB are audible; this corresponds to peaks of 1 dB.

Besides this mixing scenario, the comb filter effect can occur in stereo-to-mono conversion. For that reason, the general stereo-to-mono-conversion is not considered as a solved task [2]. Comb-filter distortions can also occur on one-microphone recordings if a direct sound wave is mixed with its reflections from wall, ceiling, floor, furniture, etc.

Practical approaches to avoid comb-filter distortions in mixing are e.g. the use of pressure zone microphones or the reduction of the number of active microphones (see [3] for details). Instead, our proposed approach is *to change the mixing process* by applying the summation of Equation (1) on short-time Fourier transformation (STFT) magnitudes and re-calculating a proper phase.

The paper is organized as follows. Section 2 introduces the concept of magnitude spectrum mixing. Section 3 shows how we can improve the phase estimation algorithm RTISI (**R**eal-**T**ime **I**terative **S**pectrogram **I**nversion) to fit better to the mixing application. Section 4 evaluates the algorithm. The paper finishes with a conclusion.

2. MAGNITUDE SPECTRUM MIXING

Figure 3 illustrates the proposed mixing algorithm.

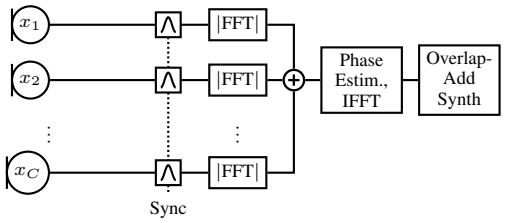

Figure 3: *Overview of the proposed mixing algorithm. FFT denotes the Fast Fourier Transform, IFFT its inverse.*

For each channel, we calculate a sequence of short-time Fourier transform (STFT) magnitudes. The STFT magnitude of the signal $x(n)$ is defined as

$$|X(mS,f)| = \left| \sum_{n=-\infty}^{\infty} x(n)w(mS-n)e^{-j2\pi fn} \right|, \quad (7)$$

where w denotes the analysis window, m the frame indices for the STFT, and S the hop size between two analysis frames. For w, we use a modified Hamming window [4]:

$$w(n) = \begin{cases} \frac{2\sqrt{S}}{\sqrt{(4a^2+2b^2)L}}(a+b\cos(2\pi\frac{n}{L})), & \text{if } 1 \le n \le L, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $a = 0.54$, and $b = -0.46$. L denotes the frame length. In the experiments, we have set $L = 4S$. The normalization factor is chosen so that

$$\sum_{m=-\infty}^{\infty} w^2(n-mS) = 1, \ \forall n. \quad (9)$$

when L is an integer multiple of $4S$. We need this property later for accurate phase estimation. Let $X_c(mS,f)$ be the spectrum series of $x_c(t)$. Then we must find a way to calculate $|X(mS,f)|$ from our single $X_c(mS,f)$ coefficients.

To find a proper formula, we concentrate on the two-channel case $C = 2$ and drop the dependence of the amplitude from the window position mS and the frequency f:

$$\begin{aligned} |X| &= |a_1X_1 + a_2X_2| \\ &= \left| a_1|X_1|e^{j\varphi_1} + a_2|X_2|e^{j\varphi_2} \right| \end{aligned} \quad (10)$$

Without loss of generality, we can set φ_1 to zero. $\Delta\varphi$ becomes the phase difference $\varphi_2 - \varphi_1$:

$$|X| = \left| a_1|X_1| + a_2|X_2|e^{j\Delta\varphi} \right| \quad (11)$$

To avoid comb filter distortions, we want to minimize the influence of the term $e^{j\Delta\varphi}$. One way to reach this goal is to set all phases to zero degrees. Then, the phase difference is also zero, and the $e^{j\Delta\varphi}$ term becomes unity. The mixing result is

$$\begin{aligned} |X| &= \left| a_1|X_1| + a_2|X_2| \right| \\ &= a_1|X_1| + a_2|X_2|. \end{aligned} \quad (12)$$

We can easily generalize these considerations to the mix of multiple input channels. This way, we can define the magnitude spectrum mixing process as the linear combination of the single channel spectrum magnitudes:

$$|X(mS,f)| = \sum_{c=1}^{C} a_c|X_c(mS,f)|. \quad (13)$$

3. PHASE RECONSTRUCTION

To reconstruct the signal from this magnitude spectrum, we need to reestimate the phase information for the STFT magnitude coefficients. For this purpose, the proposed mixing algorithm uses the RTISI method with look-ahead [5] due to its realtime capabilities and high reconstruction quality. Using the time-domain mix from Equation (1) as initial phase estimation improves the RTISI phase estimator additionally.

3.1. Analyzing and Synthesizing Audio Data

Our algorithm uses the overlap-add method [6] to reconstruct the mixed audio data. As explained in Section 2, the original audio data are split up into overlapping frames with a block size of L samples and a hop size (starting point distance between adjacent frames) of $S = L/4$ samples. The phase estimator has a look-ahead of k frames, i.e. whenever frame m is analyzed, frame $m-k$ is committed to the overlap-add synthesizer. In our setup, k is set to 3.

3.2. The Phase Estimation Buffer

The central data structure of the phase estimator is a two-dimensional buffer, which is illustrated in Figure 4. The buffer has $R = \frac{L}{S}+k$ rows. Each row has $L+(R-1)S$ elements, arranged in cells of size S.

The buffer rows store the windowed audio data for subsequent frames. Each row stores only one frame, the remaining cells are filled with zeros. The frame data are windowed with $w^2(t)$ to

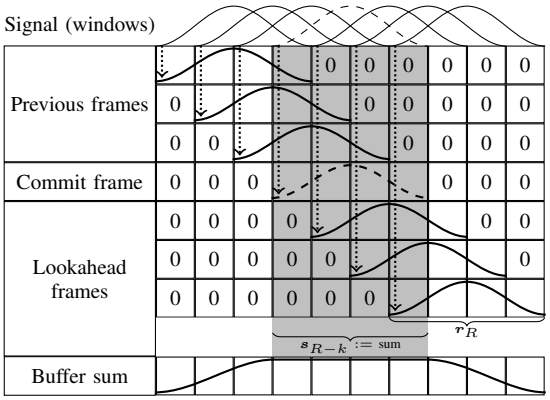

Figure 4: *Phase estimation buffer. Every cell contains S elements.*

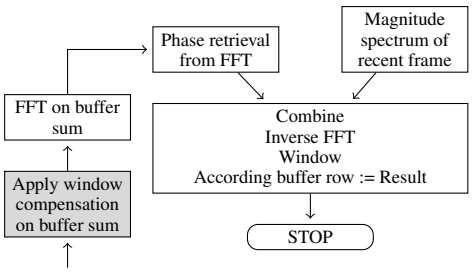

Figure 5: *RTISI estimation of one row. Modifications are drawn in light gray.*

fulfill Equation (9). If m is the frame to commit, the last row stores the frame $m+k$. The non-zero cells are arranged such that, given a fixed column, the samples in each row are synchronous.

In the following, we denote with $\boldsymbol{r}_r$ the audio data vector stored in row r (time domain). The zero cells are not taken into account. We denote the complete row vector including the zero cells as $\hat{\boldsymbol{r}}_r$. [1]

Additionally, we define the buffer sum function as the projection of the complete row vector sum to the non-zero elements according to a given row index:

$$\boldsymbol{s}_r = \left[\sum_{i=1}^{R} \hat{\boldsymbol{r}}_i\right]_{(r-1)S+1,\cdots,(r-1)S+L} \tag{14}$$

3.3. M-Constrained Transforms

The central function of the phase estimator is the *M-constrained transform*, which generates a new (and in almost cases, better) phase estimate from a given one. It operates on a given row r of the estimation buffer and is basically a five-step method (see also Figure 5). Let M be the magnitude spectrum of the frame associated to $\boldsymbol{r}_r$, obtained from Equation (13). Then, following steps are processed:

[1] We do not use any matrix algebra in this paper. All variables written in boldface are vectors. Letters with hat (e.g. $\hat{\boldsymbol{r}}$) denote vectors with the dimensionality of the full buffer row, including zeros. Accent-less lowercase letters denote vectors with L elements.

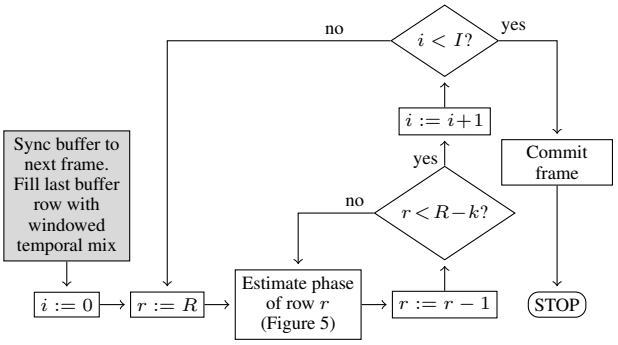

Figure 6: *The modified RTISI algorithm. Modifications are drawn in light gray.*

1. $\boldsymbol{r}_r := \boldsymbol{r}_r \cdot \boldsymbol{w}_r^*$ (element-wise, see Equation (15)),
2. $\boldsymbol{s}_r :=$ result from Equation (14),
3. $\boldsymbol{x} :=$ FFT($\boldsymbol{s}_r$),
4. $\boldsymbol{\varphi} :=$ arg $(\boldsymbol{x})$ (element-wise),
5. $\boldsymbol{x}_{\text{new}} := M \cdot e^{j\boldsymbol{\varphi}}$ (element-wise),
6. $\boldsymbol{r}_r :=$ IFFT$(\boldsymbol{x}_{\text{new}})$.

The first step is a new contribution of this paper and thus needs some explanations. Since Equation (9) contains an infinite sum and does not hold for a finite buffer (illustrated in the window sum of Figure 4), the sum of the buffer rows does not contain the actual audio data, even if the temporal mix is identical to the desired output mix. For that reason, the given magnitude spectrum does not necessarily match the sum signal. As a result, the phases are not estimated optimally. There is a partial solution for this issue presented in [5], but in the mixing application we also know the window the magnitude spectra are produced with. Thus we can compensate the effect by applying the inverse of the squared window sum on the frame and re-windowing the result with a Hamming window.

Let $\boldsymbol{w} = [w(n)]_{1\le n\le L}$ be a vector containing the non-zero values of the window function $w(n)$ from Equation (8). Assuming that each buffer row is filled with the squared window function ($\boldsymbol{r}_r = \boldsymbol{w}^2$ for each r), we can calculate the resulting window compensation function $\boldsymbol{w}_r^*$ as follows:

$$\boldsymbol{w}_r^* = \frac{\boldsymbol{w}}{\boldsymbol{s}_r} \text{ (element-wise)} \tag{15}$$

Now, for each buffer content and each row r, the inverse-windowed row signal $\boldsymbol{r}_r^* = \boldsymbol{s}_r \cdot \boldsymbol{w}_r^*$ contains the frame signal as if it had been windowed with a scaled Hamming function. Since we use the same scaled Hamming function to generate the spectrograms as introduced in Equation (7), we have matched the rows according to the magnitude spectra.

3.4. Frame Initialization

The actual frame processing is illustrated in Figure 6. Let us assume that a new frame m is processed. The first step is to synchronize the buffer to the new frame so that $\boldsymbol{r}_{R-1}$ contains the audio data of frame $m-1$, and the final row $\boldsymbol{r}_R$ is empty. For the frame m, the phase estimator gets following information: the magnitude

spectrogram mix $|X(mS, l)|$ from Equation (13), and the temporal mix $x(t)$ from Equation (1).

After buffer synchronization, the phase estimator windows the temporal mix with $w^2(t)$ (to fulfill Equation (9)) and stores it into $\boldsymbol{r}_R$. This step forces the phase estimator to use the phase of the additive mix as initial phase for the output and thus provides a better initial phase estimate than the original RTISI estimator gets.

3.5. Transform Iterations and Look-Ahead

After buffer initialization, we apply the M-constrained transform iteration as described in section 3.3 on $\boldsymbol{r}_R$. Then, we apply this iteration on the preceding rows according to Figure 6 until we have reached $\boldsymbol{r}_{R-k}$. We repeat the whole iteration sequence several times. Finally, we commit $\boldsymbol{r}_{R-k}$ to the overlap-add synthesizer. As described in [5], the advantage of a look-ahead like this is that we have some knowledge about future frames before we finalize a frame's phase estimation and commit the frame.

4. EVALUATION

To compare the proposed method with the temporal mix, it is important to use a proper criterion.

A perceptual measure seems much more valid for this task than signal-theoretic methods such like magnitude spectrogram signal-to-error ratio (SER, [5]) for the following reason: If the input signals do not contain time-delayed versions of the same source, the mixer output should *sound exactly like* the output of the time-delayed mix. If one channel signal contains a time-delayed version of another channel's signal, the mixer output should *sound as close as possible* as the input signal of one channel. In both cases, an accurate perception is more important than a high SER.

The recent standard for perceptual audio quality evaluation is ITU-R BS.1387-1, also called PEAQ (**P**erceptual **E**valuation of **A**udio **Q**uality, [7]). In this paper, the EAQUAL implementation [8] of the PEAQ basic model is used.

4.1. Test Setup

The PEAQ Objective Difference Grade (ODG) compares two signals, namely a reference signal and a (degradated) test signal. The test setup is illustrated in Figure 7. We take one original signal, delay it by a given amount of time T, and mix the original signal with the delay either in the time domain or using the proposed method. PEAQ now compares the original and the mix signal. The mix signal is normalized such that its energy equals the original signal's energy.

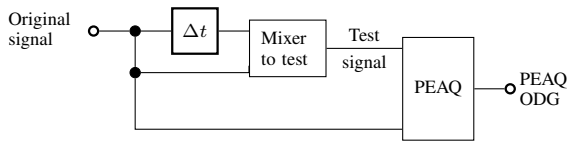

Figure 7: *Test setup for evaluating mixing algorithms.*

As reference material, we have chosen the beginning of Stefan Raab's song "Hier kommt die Maus". Instrument examples (organ, cello) from the EBU SQAM library [9] have led to similar results.

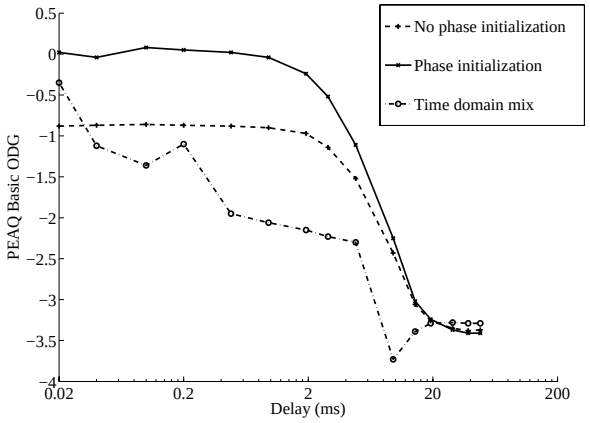

Figure 8: *PEAQ Objective Difference Grades vs delays. L=2048, S=512, I=10*

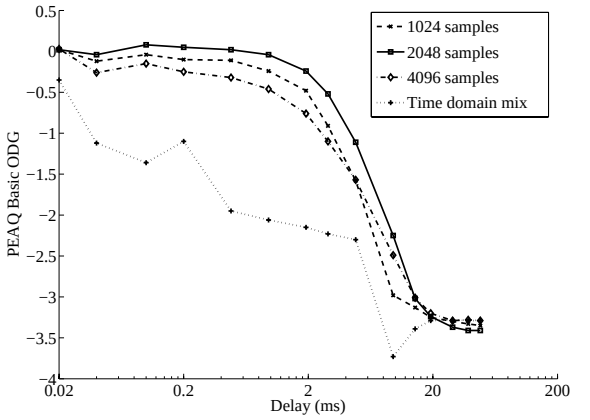

Figure 9: *PEAQ Objective Difference Grades vs. delays with different window sizes L. S=L/4, I=10. The phase estimator is initialized with the temporal mix.*

4.2. General Results

Given appropiate parameter settings, we can say that the proposed method outperforms the temporal mix in terms of PEAQ ODGs for delays lower than 10 ms. For delays lower than 2 ms, the ODG values are above -1 (which stands for "perceptible, but not annoying"). See Figure 8 for details.

For interest, we have also included the results generated without using the initial estimation from the temporal mix. We can see that the initial estimation improves the result by nearly one PEAQ difference grade. A possible reason is that the phase of most frequency bands is given more accurate in the temporal mix than a blind estimation from magnitude spectrograms delivers.

Nevertheless, these results should be interpreted with great care because the PEAQ measure is designed for high-quality audio comparison. For lower quality grades, other measures can predict the results of psychoacoustical experiments better [10]. For that

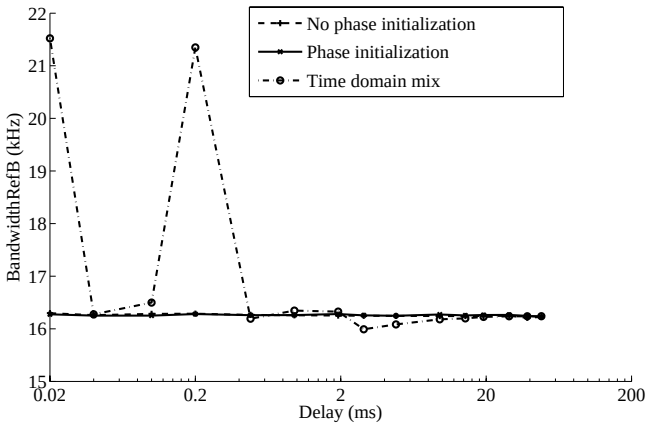

Figure 10: *PEAQ* BandwidthRefB *model output variable vs delays.*

reason, we should not overestimate the accuracy of differences in low PEAQ values, which occur on high delay times in any configuration.

To understand the outlier in the temporal mix at 10 samples delay (at 48 kHz sampling rate; i.e. ca. 0.2 ms), we must recall that the ODG value is calculated from multiple model output variables (MOVs). Two MOVs also have this outlier: *BandwidthRefB* and *BandwidthTestB*. As stated in [11], PEAQ defines the bandwidth as the frequency bin which amplitude exceeds the high-frequency maximum by 5 dB (*BandwidthTestB*) or 10 dB (*BandwidthRefB*). The high-frequency maximum is defined as the maximum amplitude of the FFT frequency bins with a frequency ≥ 21.6 kHz.

Now, in the case of 10 samples, the comb filter creates a frequency notch at exactly 21.6 kHz (see Equation (6), k=4). Consequently, the amplitudes of these frequency bins are especially low; so the high-frequency maximum becomes low, resulting in a low amplitude threshold to determine the bandwidth. As a result, the bandwidth for this delay seems higher (see Figure 10 for the *BandwidthRefB* case).

4.3. Window Size and Transform Iterations

Evaluating different window sizes L with the overlap factor $\frac{L}{S} = 4$ kept constant, we can see that a window size of 2048 can be considered to give best results. See Figure 9 for details.

Evaluating different numbers of transform iterations I, it can be shown that the number of iterations has little influence on the ODG.

5. CONCLUSIONS AND OUTLOOK

In this paper, a novel approach to audio mixing is presented which is capable to avoid comb-filter distortions while having only a very small degradation when mixing signals without time delays. Compared with mixing in the time domain, the drawbacks of this algorithm are the latency due to the buffering and the look-ahead, and the computational complexity.

Future resarch may include a broader evaluation using the advanced model of PEAQ and psychoacoustical hearing tests. This holds especially for setups with delays longer than a few milliseconds. The evaluation may also include mixing scenarios with multiple sources. First experiments with multiple sources are currently work in progress.

6. REFERENCES

[1] S. Brunner, H.-J. Maempel, and S. Weinzierl, "On the Audability of Comb Filter Distortions," in *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007, No. 7047.

[2] V. Välimäki, S. Gonzáles, O. Kimmelma, and J. Parviainen, "Digital audio antiquing — signal processing methods for imitating the sound quality of historical recordings," *Journal of the Audio Engineering Society*, vol. 56, no. 3, pp. 115–139, March 2008.

[3] G. Ballou, *Handbook for Sound Engineers*, pp. 424,475,609f, Focal Press, 3 edition, 2005.

[4] D. Griffin and J. Lim, "Signal Estimation From Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[5] X. Zhu, G. Beauregard, and L. Wyse, "Real-Time Signal Estimation From Modified Short-Time Fourier Transform Magnitude Spectra," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1645–1653, 2007.

[6] J. Allen and L. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[7] ITU-R Recommendation BS.1387-1, *Methods for Objective Measurements of Perceived Audio Quality*, 2001.

[8] A. Lerch, "EAQUAL, Version 0.1.3," http://www.mp3-tech.org/ programmer/sources/eaqual.tgz.

[9] European Broadcasting Union, "Sound Quality Assessment Material," Tech 3253, 1988, http://www.ebu.ch/en/technical/publications/tech3000_series/tech3253/.

[10] C. Creusere, K. Kallakuri, and R. Vanam, "An objective metric of human subjective audio quality optimized for a wide range of audio fidelities," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 129–136, 2008.

[11] P. Kabal, "An Examination and Interpretation of ITU-R BS. 1387: Perceptual Evaluation of Audio Quality," *McGill University*, 2002.