Seamless Acoustics Matching of Disparate Recordings

Ramin Anushiravani
for Submission
City, Country
ramin.audio@gmail.com

Paris Smaragdis
for Submission
City, Country
e-mail address

ABSTRACT

This paper presents an intelligent DAW interface for matching acoustical characteristics of multiple audio recordings, an operation that is crucial when splicing multiple audio recordings into one. The matching process ensures a continuity in sound characteristics, the lack of which is often very distracting to a listener. Unfortunately, the complexity of manually editing the acoustical features of recordings makes it difficult for users to perform such a task. The proposed approach automates the acoustic matching procedure by estimating acoustical features that are contained in an example sound and applying them to a desired recording. When the proposed algorithms were used as an acoustics matching interface, it is shown that it produces better results than manual processing by the users, and produces results significantly faster without necessitating audio processing expertise.

ACM Classification Keywords

H.5.5[Multimedia Information Systems]: Sound and Music Computing, H.5.2 [User Interfaces]: Voice I/O

Author Keywords

Acoustics matching, Audio interfaces, Example-based editing

INTRODUCTION

Digital audio workstations (DAWs), such as Adobe Audition and Audacity enable users to perform complex audio editing operations using basic signal processing and machine learning building blocks. Most DAWs, however, require familiarity with audio processing and strong listening skills. Their current interface format spans back to the first digital editors of the 70s, which relied heavily on tape editing processes. Unfortunately these interfaces are often too complex for the casual user, necessitating a solid understanding of acoustics, audio processing terminology, and the ability to identify various acoustic artifacts and map them to a processing step that can potentially ameliorate them. Even in the case of expert users, the necessary editing is too complex to be done manually (similarly to how modern image processing tasks were intractable before the interface innovations brought to us by modern image editors).

Every submission will be assigned their own unique DOI string to be included here.

Content creators often record sound clips using different microphones, in different rooms, and with different background noises in their respective environments. When splicing these recordings together, the lack of consistency between the recordings can be distracting for the listener. It is up to the content creator to manually adjust the acoustics of each recording using a DAW. This places a significant burden on a user who would benefit from a simplified interface for such a task.

Another way to make an acoustically consistent combined recording is to process all the individual recordings by flattening their equalization, removing the noise, and performing dereverberation [8, 7, 13, 16] on them. Such drastic techniques, however, introduce significant audible artifacts, that often make an unnatural sounding output which might be more upsetting to the listeners than simply having mismatched recordings. We argue that acoustic matching can do a better job in making consistent recordings by matching the existing equalization, noise, and reverberation.

In the following sections we will briefly introduce some of the computational steps that are required to perform acoustic matching, and then we will present a system that incorporates them.

EQUALIZATION MATCHING

Equalization expresses the gain of a recording at different frequencies. In current DAWs, adjusting the equalization of a recording is usually done through a graphical interface that consists of a number of bars for adjusting the gain of the recording at selected frequencies [3].

In order to automatically match the equalization of a recording, we propose, first to calculate the power spectras (i.e. energy of the sound at different frequencies) of the recordings [18]. We then derive an "equalization filter" defined as:

$$E(k) = \frac{P_{ex}(k)}{P_{in}(k) + \beta(k)} \tag{1}$$

where $(.)_{in}$, $(.)_{ex}$, $P(.)$, k, and E denote input, example, power spectra, frequency bin, and the equalization filter, respectively. $\beta(k)$ is a frequency-dependent regularization parameter to avoid ill-conditioned frequencies [5]. The resulting filter is then applied on the input sound in the frequency domain. This is simply done by element-wise multiplying the equalization filter with the input sound power spectrogram (i.e. squared of the time-frequency representation of a sound) as [21]:

$$Y_{mat}[t,k]^2 = E(k)X_{in}[t,k]^2 \tag{2}$$

where $(.)_{mat}$ denotes the matched sound and t denotes time frame index. Doing so ensures that the power spectrum of the resulting recording will have the same power spectrum as the target sound. Note that the equalization filter can be more easily applied using time-domain filtering, but for simplicity we use an STFT implementation since this is the domain in which we perform the other two matching processes.

NOISE MATCHING

Background noises are common to recording sound clips. Noise reduction toolboxes in most DAWs usually involve capturing a "noise profile" and then subtracting that from the noisy recording through fine-tuning a number of parameters [4].

A noisy recording can be represented as [16]:

$$Y[t,k] = X[t,k] + V[t,k] \tag{3}$$

where Y, X, and V denote the magnitude spectrogram of the noisy, clean, and noise recordings respectively. The first step in automatically matching the noise of the input sound to that in an example sound is to extract the noise and the clean sounds from both recordings. We used Spectral Subtraction [6] to decompose the noisy recordings into the noise and clean components. After extracting the noise profile for each recording, the equalization matching from the previous section (denoted as EQ) can be used to match the background noise as shown in equation 4.

$$V_{mat} = \mathrm{EQ}(V_{in}, V_{ex}) \tag{4}$$

The equalized noise is then added as follows to the input recording using the appropriate signal-to-noise ratio (SNR) as estimated from the example sound [23].

$$Y_{mat}[t,k] = X_{in}[t,k] + \alpha V_{mat}[t,k] \tag{5}$$

Where α is the noise gain corresponding to the estimated SNR.

REVERBERATION MATCHING

Reverberation (in short reverb kernel) is the weighted averages of the reflection of the sounds off of walls and objects in a room before reaching a listener. Most DAWs have tools for reverberating an audio clip through adjusting parameters like the room size and the location of the source in the room, etc. A few DAWs also attempt to reduce the reverberation in a recording by using deconvolution techniques [9], although the results are often not high-quality and requires many input parameters from the user.

In reverberation matching, the goal is to change the room effect in a recording to resemble the effect of a different room (e.g. editing a sound recorded in a small room as if it was recorded in a concert hall). A reverberant sound can be expressed as the linear combination of the decayed and delayed copies of the dry (i.e. without reverb) sound. We used equation 6 to synthesize reverb kernels in this paper [10].

$$h[n] = b[n]e^{-\zeta n} \tag{6}$$

where bs are coefficients for a zero-mean Gaussian noise and ζ controls the decay rate of the kernel. Reverberation can be approximated as the convolution between the input and the kernel spectra [12, 15, 17, 22, 11, 2] as:

$$Y \approx X \star L \tag{7}$$

The variables X, Y, and L represent the magnitude spectrogram of the dry sound, reverb sound, and the kernel, respectively. The operator $\star$ denotes convolution between corresponding rows of X and L at each frequency bin for all time frames.

We proposed a convolutive non-negative matrix factorization (CNMF) [20] scheme, which is an extension to non-negative matrix factorization (NMF) [14] that decomposes a reverberated sound into a dry sound and a reverb kernel[1]. We then reconstruct the reverb input recording using dry input sound and the example kernel. In summary, we performed the following steps as also shown in Fig. 1. The first three steps are applied to both input and example sounds.

1. Estimate dry speech bases (W_d) by decomposing the magnitude spectrogram of dry speech recordings (Y_d) using NMF.

   $$(W_d, H_d) = \mathrm{NMF}(Y_d, \alpha) \tag{8}$$

   where H_d and α denote the corresponding dry activation matrix and a sparsity parameter enforced on the estimated basis, respectively.

2. Estimate the reverberated activation matrix (H_r) by decomposing the magnitude spectrogram of the reverberant recording (Y) using CNMF for both input and example sounds.

   $$H_r = \mathrm{CNMF}(Y, W_d, \alpha) \tag{9}$$

3. Estimate the example kernel and the input dry activation matrix. This is done by decomposing the corresponding H_r by enforcing a sparsity parameter (β) on the dry activation matrix (H_d).

   $$(H_d, R) = \mathrm{CNMF}(H_r, \beta) \tag{10}$$

4. Construct the matched sound using a matched kernel to estimate the reference kernel through exponentiation of the estimated input kernel.

   $$Y_{mat} \approx W_d \cdot (H_{d,in} \star R_{mat}) \tag{11}$$

USER STUDY

We conducted a user study to understand if the proposed system is useful in increasing the efficiency of manually editing the acoustics of recordings, and whether it can achieve higher quality results. We interviewed 31 people with different backgrounds in audio editing. Users were categorized into three categories: experts, moderately experienced, and novice users (approximately 10-12 people in each category). This selection was made through an initial interview with the user and their previous skills in music production and audio editing. We used a standard interface for manually editing the equalization and reverberation of a recording, which included a parametric equalizer and a simple slider to adjust reverberation time. In contrast, our proposed system simply required the selection of two files to match.

Each user was asked to perform 15 tasks. The goal of each task is to modify an input recording through the provided tools,

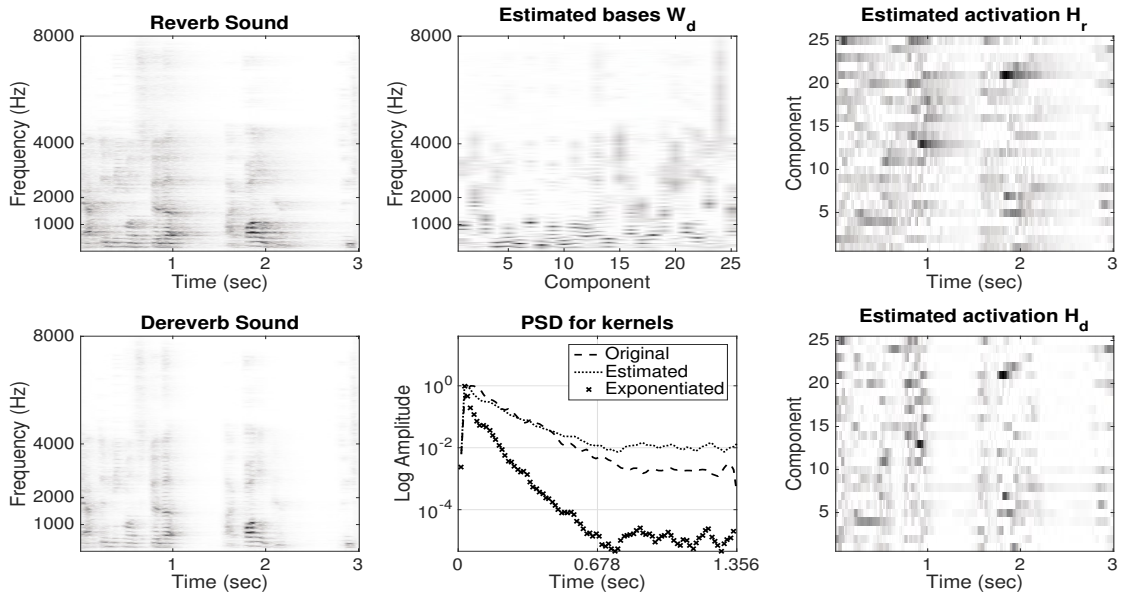


Figure 1: Example decomposition and reverb modification. The depicted reverb sound is a 10 seconds long recording recorded in a big lecture hall. We zoomed in on the first 200 STFT frames for visual clarity. H_r was estimated by decomposing the reverberant speech sound using equation 9. CNMF is then applied on H_r to estimate H_d and R using equation 10. For illustrative purposes, R was averaged over all components. We reconstructed a less reverberated sound by setting $\beta = 2.5$ in equation 10. For reverb matching with an example sound, we would estimate β from a another recording so that we can match the amount of reverberation

first manually, then using the acoustics matching system, to acquire a desired feature in the provided example sound. Each recording was a five-second long speech recording sampled at 16 kHz. At the end of each task, the user was asked to fill out a form with questions on the complexity and frustration of the manual editing, and their confidence in achieving the correct result. Upon completing the acoustic matching tasks using the proposed system, the user was also asked to evaluate the quality and the efficiency for the same recording. For all these questions we used a scale from 0 to 5.

TASKS

Task I

In the first task, users were asked to perform equalization matching. We changed the gain of the input sound by 16 dB, 10 dB, and 4 dB in different frequency regions, resulting in what we term as easy, medium, and hard cases to perform matching. These gains were applied to each of the these frequency regions: $f < 500$ Hz, $f > 2000$ Hz, and 1000 Hz $< f < 4000$ Hz (9 recordings in total). The example sounds had a natural spectra. Users were asked to equalize the input sound so that it matches the spectra of the example sound. Users first started with the boosted low frequency recordings, starting from 16 dB (easy cases) down to 4dB (hard cases), and then followed the same procedure for the high-frequency and mid-frequency affected recordings. Users were not told in advance what the gains, or the frequency ranges were; they were supposed to listen to the results and judge whether they were performing the right operations.

Task II

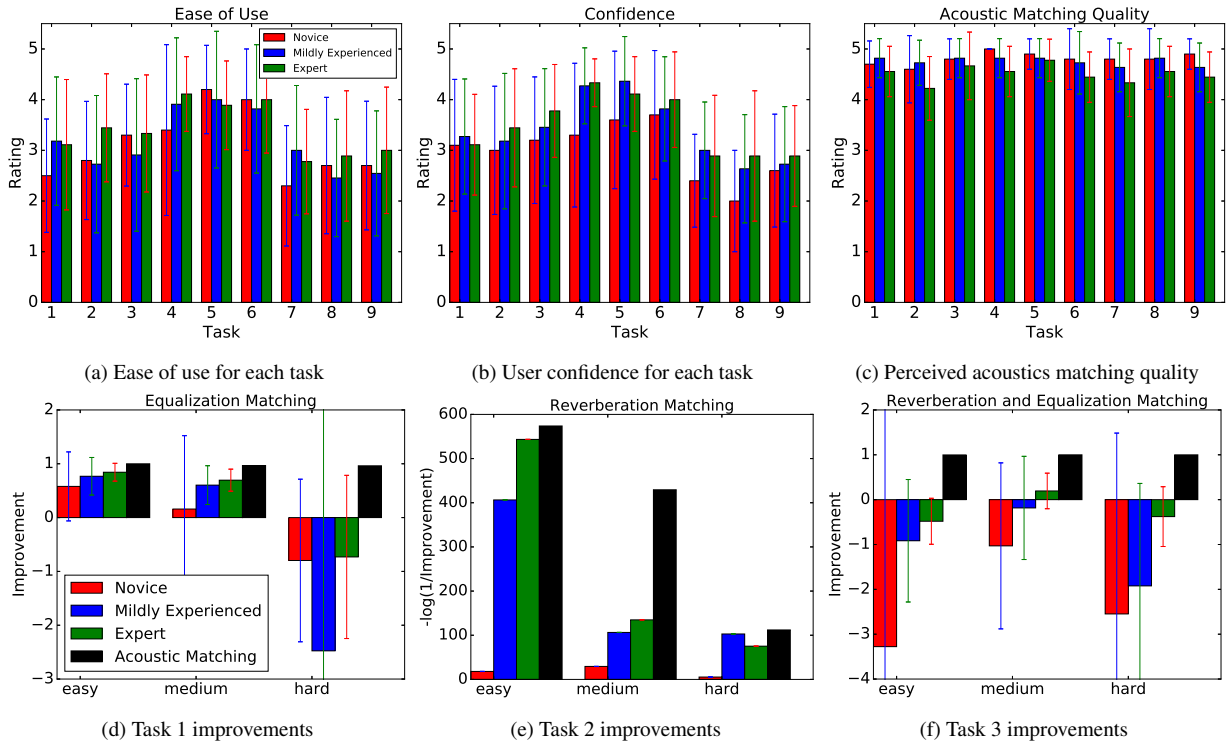
The goal of this task was to add reverberation to the input recordings until it contained the same amount as the provided example sound. Users were asked to adjust the parameter ζ from equation 6, which modifies the slope of the reverb kernel. We synthesized a kernel based on this parameter and convolved it with the input recording to reverberate the sound. Example recordings were reverberated in the same manner using a 0.8 seconds long reverb kernel with $\zeta = [0.25, 0.91, 1.73]$ which corresponds to easy, medium, and hard matching cases, respectively (e.g., 3 recordings in total).

Task III

The purpose of this task was to test the users on a more realistic situation, where both the equalization and reverberation needed to be adjusted. Input recordings had no equalization or reverberation. The users were asked to equalize and reverberate them until they matched the presented example sounds. The example recordings were boosted by 10 dB in the low, high, and bandpass frequency regions (as described in Task I) with $\zeta = 0.6$ corresponding to easy, medium, and hard matching cases (3 recordings total).

RESULTS

We evaluated the results of the study based on the user's performance and their responses to a questionnaire. Figures 2.a and 2.b show the ease of use and confidence for performing each task manually. Most users evaluated of the complexity of each task similarly. A user's experience in audio processing seemed to be correlated with more confidence in achieving



(a) Ease of use for each task

(b) User confidence for each task

(c) Perceived acoustics matching quality

(d) Task 1 improvements

(e) Task 2 improvements

(f) Task 3 improvements

Figure 2: Results from the user questionnaire and using numerical evaluation of the acoustics matching.

the correct recording for most of the tasks. Users also seem to find reverberation matching less complex and frustrating than the other tasks, resulting in a higher confidence as compared to the other tasks (this is perhaps due to the simplification of the manual editor to only one parameter). The final task seems to be the most frustrating and complex, exhibiting the lowest confidence scores. Many users expressed that the reason for lower confidence was because when they were adjusting the equalization they noted their edited sound didn't seem to have the amount of reverb before they had equalized the sound. Same thing for changing the reverb, it had changed the equalization as well. This perceived correlation between the tools had made this task more challenging. Another reason could also be the increase in the number of parameters.

Figure 2.c shows the perceived quality of the proposed acoustic matching system. All users agreed that the automatic acoustics matching system not only exhibits higher quality results, but that it was also more efficient as compared to the manual editing. Experts seem to rate the resulting quality slightly lower than the non-expert users. We believe that these lower ratings are due to subtle artifacts that are introduced by the algorithm. For example, the reverb matching algorithm would have a harder time analyzing a recording when the amount of reverb is very low. Also, since the reverberant signal phase is used to reconstruct the recordings some traces of the original reverberation can often persist. Expert users seemed to be more sensitive to such artifacts.

We also show the improvement scores by calculating the difference between the Kullback-Leibler distance (KLD) between the normalized power spectra of the recordings for task 1 and 3 (figures 2.d and 2.f), and the normalized power spectra of the synthesized kernel for task 2 (figures 2.e) [19]. These measures produce a more objective measurement of the quality of matching.

$$\text{improvement} = \frac{D(\text{in},\text{ex}) - D(\text{mat},\text{ex})}{D(\text{in},\text{ex})} \tag{12}$$

where D denotes the KLD distance. We show the improvement score for completing each task manually for each category, as well as for the acoustic matching score. As expected, all users performed better when the effect was more audible or less complex to work with (i.e. required a few parameters adjustment). More experienced users performed better when matching the reverberation, but struggled when matching both the equalization and reverberation. Experts seem to be performing better than other users for all tasks. The acoustic matching system outperformed all users in all tasks.

CONCLUSIONS

We presented a system that helps users to efficiently and accurately perform acoustic matching between recordings. We designed a system that automates the acoustics matching process, and for simple tasks decreased user's frustration and improved user's satisfaction. The tasks used in this paper were simplified cases such that we could gather meaningful results from users. In real-life recordings only highly-trained experts would stand a chance to match the performance of this system, but only after a significant investment of time.

REFERENCES

1. Ramin Anushiravan. 2016. Example-Based Audio Editing. Master's Thesis. (2016). University of Illinois at Urbana Champaign.
2. Ramin Anushiravani, Paris Smaragdis, and Gautham J. Mysore. 2015. Long Reverberation Matching. (2015).
3. Adobe Audition. 2015a. Filter and equalizer effects. Video. (2015).
4. Adobe Audition. 2015b. Noise reduction techniques and restoration effects for Audition. Video. (2015). http://tv.adobe.com/watch/companion-videos-for-inspire/adobe-audition-cc-removing-noise/
5. Edgar Y. Choueiri. 2010. Optimal Crosstalk Cancellation for Binaural Audio with Two Loudspeakers. Princeton University.
6. Thomas Esch and Peter Vary. 2009. Efficient Musical Noise Suppression For Speech Enhancement Systems. In *Institute of Communication Systems and Data Processing RWTH Aachen University, Germany*. IEEE.
7. Kun Han, Yuxuan Wang, DeLiang Wang, William S. Woods, Ivo Merks, and Tao Zhang. 2015. Learning Spectral Mapping for Speech Dereverberation and Denoising. textupin ACM Transaction on Audio, Speech,and Language Processing.
8. iZotope. 2013. Principles of Equalization. Tips and Tutorials - Online. (December 2013).
9. iZotope. 2016. Reducing Reverb with the RX De-Reverb Module. iZotope Support-Online. (2016).
10. Jean-Marc Jot, Laurent Cerveau, and Olivier Warusfel. Analysis and Synthesis of Room Reverberation based on a Statistical Time-Frequency Model. in Institute for Research and Coordination in Acoustics/Music.
11. Ante Jukic, Toon van Waterschoot, Timo Gerkmann, and Simon Doclo. 2014. SPEECH Derverberation with Convolutive Transfer Function Approximation Using MAP AND Variational Deconvolution Approaches. in International Workshop on Acoustic Signal Enhancement.
12. Kshitiz Kumar. 2011. A Spectro-Temporal Framework for Compensation of Reverberation for Speech Recognition. In *PhD thesis*. CMU.
13. Kshitiz Kumar, Bhiksha Raj, Rita Singh, and Richard M. Stern. 2011. An Iterative Least-Squares Technique for Dereverberation. International Conference on Acoustics, Speech and Signal Processing.
14. D.D Lee and H.S. Seung. 2000. Algorithms for Non-Negative Matrix Factorization. In *Advances in Neural Information Processing Systems*, Vol. 13. NIPS.
15. Dawen Liang, Matthew D. Hoffman, and Gautham J. Mysore. 2015. Speech Dereverberation using a Learned Speech Model. in International Conference on Acoustics, Speech and Signal Processing.
16. Philipos C. Loizou. 2013. Speech Enhancement: Theory and Practice, Second Edition. CRC Press.
17. Nasser Mohammadiha, Paris Smaragdis, and Simon Doclo. 2015. Joint Acoustic and Spectral Modelling FOR Speech Derverberation Using Non-Negative Representations. in International Conference on Acoustics, Speech and Signal Processing.
18. Alan V. Oppenheim and George C. Verghese. 2010. Power Spectral Density. In *Introduction to control and signal processing*.
19. David Pinto. 2007. The Kullback-Leibler Distance. in International Conference on Intelligent Text Processing and Computational Linguistics.
20. Paris Smaragdis. 2007. Convolutive Speech Bases and their Application to Supervised Speech Separation. In in *Speech And Audio Processing*. IEEE.
21. J.O. Smith. 2011. The Short-Time Fourier Transform (STFT) and Time-Frequency Displays. In *Spectral Audio Signal Processing*. https://ccrma.stanford.edu/~jos/sasp/Mathematical_Definition_STFT.html#19930
22. R. Talmon, I. Cohen, and S. Gannot. 2009. Relative Transfer Function Identification using Convolutive Transfer Function Approximation, Vol. 17. Audio, Speech, and Language Process, 546 – 555.
23. Martin Vondrase and Petr Pollak. 2005. Methods for Speech SNR estimation: Evaluation Tool and Analysis of VAD Dependency. Radio Engineering.