Audealize: Crowdsourced Audio Production Tools

Prem Seetharaman AND Bryan Pardo
(prem@u.northwestern.edu) (pardo@cs.northwestern.edu)

Northwestern University, Evanston, US

Audio production involves using tools such as reverberators and equalizers to transform audio into a state ready for release. While these tools are widely available to musicians who are not expert audio engineers, existing interfaces can be frustrating for newcomers, as their interfaces are parameterized in terms of low-level signal manipulations that may not be intuitive to non-experts. We present Audealize, an interface that bridges the gap between low-level parameters of existing audio production tools and programmatic goals (e.g. "make my guitar sound 'underwater"'). Users modify the audio by selecting descriptive terms in a word-map built from a crowdsourced vocabulary of word labels for audio effects. We perform the first user study comparing a word-map interface to traditional audio production interfaces. A study on a population of 432 non-experts found they favored the crowdsourced word-map over traditional interfaces. Absolute performance measures show those using the word-map interface produced results that equaled or exceeded results using traditional interfaces. This indicates language, in concert with a meaningful word-map visualization, is an effective interaction paradigm for audio production by non-experts. Audealize is an example of a general design approach that can apply to other domains and can be accessed at http://audealize.appspot.com.

0 Introduction

Audio production involves using tools such as reverberators, equalizers and compressors to make audio for music, video, podcasts, etc. ready for public consumption. In recent years, high quality audio production software, once found only in professional recording studios, has become affordable and available to a broad range of people. These people wish to use production tools to achieve artistic effects, but are typically not professional audio engineers.

Artistic creators often want to use production tools to evoke extra-musical impressions in an audience. For example, the beginning of Pink Floyd's "Wish You Were Here" evokes the effect of a guitar being played along with music over a tinny radio. The lead guitarist for the band, David Gilmour, did not play his guitar through a radio, but rather used audio production tools to manipulate his guitar sound to match that of a car radio [1]. We call this *programmatic music production*, after the classical music tradition of *programmatic music*, where composers try to evoke extra-musical thoughts in their audience (e.g. Prokifiev's *Peter and the Wolf* evoking distinct animals through orchestration and melody choices).

Current production tools, such as equalizers (Figure 1) and reverberators (Figure 2), are not designed to naturally support programmatic music production. Their interfaces are conceptualized in a *signal-parameter space*, which consists of low-level technical parameters of the signal processing techniques behind the audio tools. While David Gilmour has experience with these interfaces for audio production tools and access to expert audio engineers to achieve their creative goals, our target population does not.

Figure 1 shows a typical parametric equalizer. It is not obvious to a non-engineer (e.g. an acoustic musician or amateur) how to manipulate the knobs and dials to achieve David Gilmour's goal of making his guitar sound like it's coming out of a transistor radio. For many, the relationship between the *programmatic space* and the *signal-parameter space* is unknown and is a major barrier that adds significant time and frustration to the process.

Expert recording engineers can navigate the signal-parameter space effectively, but often do not share a common descriptive language with non-engineers. As a well known engineer put it: "It's a situation all engineers have been in, where a musician is frustratedly trying to explain to you the sound he or she is after, but lacking your ability to describe it in terms that relate to technology..." [2]. Further, inspiration may happen at times or places (e.g. midnight in a singer-songwriter's basement) when it is not possible to access an expert engineer. These availability and communication issues mean hiring an expert engineer may not solve the problem.

Developers of commercial audio production tools [3] often try to simplify the interfaces with a list of named preset parameter combinations (presets). Popular commercial

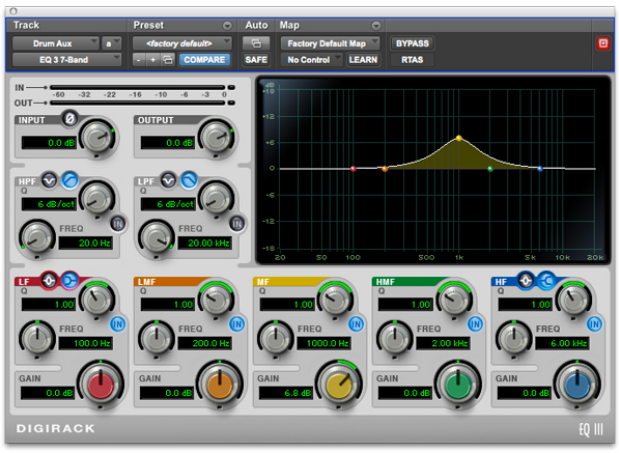


Fig. 1. A parametric equalizer. The large number of knobs on the interface can be intimidating to a newcomer. How would you manipulate this interface to make something sound "tinny"?

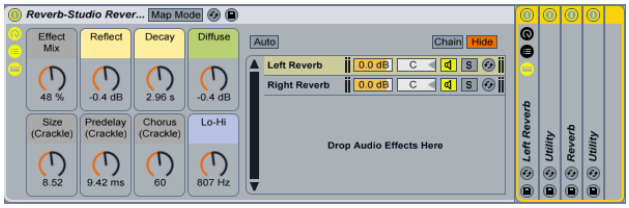


Fig. 2. A parametric reverberator from Ableton Live, a digital audio workstation. For a novice, it isn't clear how you would use this interface to make something sound "warm" or "underwater".

tools (e.g. Adobe Audition, Ableton Live) often have hundreds of presets in multi-layer menus, making the preset list nearly as difficult to navigate as the original interface. Further, lists do not show relationships between presets and are often named using language that those who did not develop the tools may not share (see Section 4).

In this work, we describe Audealize, an interface that lets the user modify sound by selecting descriptive terms (e.g. "underwater", "tinny", "chaotic") in a 2-D word-map (Figures 3 and 4), where spatial relationships between words indicate relatedness of the audio effects. When a word is selected, the sound is modified to embody the descriptive term. The interface vocabulary is seeded from a crowdsourced set of natural language descriptors collected in two previous studies [4] [5], rather than a curated list made by tool-builders or experts in audio (e.g. audiologists, recording engineers, developers of audio plugin software). This is a deliberate choice. Different communities may use different language to describe the same things, therefore we feel it best to learn the appropriate vocabulary for a particular population from that population. To build a version of Audealize for a target population (e.g. hobbyists, orchestral musicians, visual artists, Spanish speakers), one would collect a vocabulary from the target population. Users may also teach new vocabulary to the tool. This lets Audealize automatically adapt to the population of users.

By working in the programmatic space, we create a consistent user experience regardless of the underlying tool. Figures 1 and 2 show very different interfaces for two audio production tools. In contrast, Audealize (Figures 3 and 4) has analogous interfaces for both reverberation and equalization. While our work focuses on audio production, this approach to interface design and construction could be applied more broadly, since the controls are in the programmatic space instead of the signal-parameter space. For example, one could imagine a Photoshop (or Instagram) tool with a crowdsourced interface that makes pictures "warmer" or "old-timey".

This publication is the first to directly compare the efficacy of a 2-D word-map to the standard interface, which we call a signal-parameter interface for the remainder of this article, for audio production tools (equalization and reverberation) via a user study. Audio effects plugins often have preset lists and these are the only readily available word vocabularies with precisely-defined settings on the relevant audio effects tools. We therefore compare crowdsourced vocabularies for equalization and reverberation vocabularies to the vocabularies created by expert tool builders and embodied in preset lists. This is the first work to create and evaluate a unified programmatic interface that spans multiple effects tools. We also describe how to make the tool a life-long learner that continues to update vocabulary in response to user interaction.

In the remainder of this paper we first place our work in context of existing work. We then show how to automatically build a programmatic word-map interface for audio production from crowdsourced data. We establish why one should use a crowdsourced vocabulary, rather than an expert-curated vocabulary. Finally, we give the results of user studies that compare the new interface to traditional interfaces on two different tasks. The first is to modify the sound with a programmatic goal in mind (e.g. make it sound "underwater"). The second is to modify the sound with a non-programmatic goal in mind (e.g. modify this recording to match an existing recording).

1 Related Work

1.1 Map-based Audio Interfaces

Several groups have explored using 2-D maps to control audio effects, mixing, sound synthesis or search through sample libraries. Cartwright [6] created a map-based interface for mixing together multiple audio tracks. This map had no labels, which users found problematic for navigation. The work in [7] put audio samples (short recordings, like a single snare drum hit) on a 2D map, where similar samples were placed closed to each other. Like Cartwright, the samples were placed on the map without any labels.

Hoffman [8] explored synthesizing sounds in a parameter space of perceptually relevant features. Labels provided by the authors were used. This can be problematic when the vocabulary does not align with the vocabulary of the user (see Section 4).

Sabin et al. [9] used four author-chosen labels ("bright", "warm", "tinny", "dark") placed on a 2-D map to control an equalizer. Mecklenburg [10] created a similar interface for equalization, using tens of terms. However, terms were

chosen and defined by the authors and mixed low level and programmatic concepts (e.g. “warm hi-cut”, “less sizzly”). Amateur musicians considered some of their terms too technical, indicating a disconnect between those who determine the vocabulary and those who used the tool. Our system avoids this issue by crowdsourcing terms from the population that will use the tools.

The most closely related work is our own preliminary work [11], which described a word-map interface for a reverberator. The work presented here is considerably more advanced. The word mapping to spatial location better reflects relationships between words, our system is language agnostic (e.g. works with both English and Spanish), we add a learning component, our interface works with multiple audio production tools, and we have evaluated the efficacy of a word-map interface for both reverberation and equalization with a large-scale user study. None of these things were done in our preliminary work.

1.2 Content Creation/Control with High-level Descriptors

There is some prior work in building audio controllers parameterized with natural language terms. Schmidt [12] made an early example of a computer-assisted audio production tool using natural language as its interaction paradigm. It had a small dictionary of expert-selected words related to a library of audio samples (e.g. recordings of flutes and tubas, etc) and production tools (e.g. MIDI sequencer). It used a language parser to let the user control system by saying things like “Play the bass”. This work focused on audio samples and sequencing rather than audio effects, did not include a learning component and did not use a visual word-map.

Mycroft and Paterson [13] described the effect that current signal-parameter equalizer interfaces have on a user’s creativity, approach, and final output. Their work described some directions for equalizer interface design. Our work, in contrast, presents and evaluates a novel interface for controlling multiple audio effects. Mo et al. [14] found that emotional cues can be conveyed via reverberation. Some of their emotional categories, such as romantic and mysterious, overlap with the vocabulary found in our data collection [4]. Their work is a vocabulary study, like our prior work [4], whereas our current work uses a vocabulary to build and test an interface.

Huang [15] made a control knob for synthesis of new sounds parameterized by a natural language word (e.g. “scary”), learned from interaction with a user. Their system was for the synthesis of sounds, rather than the application of audio effects. The labels are also provided by a single user, whereas our labels are provided and verified by hundreds of people.

Stables et al [16] presented tools that can be used directly in a digital audio workstation. Data was collected from people using the tool. Settings of the audio tools are related to descriptors provided by the user. Users could achieve the sound they sought by providing a natural language a description. Our work leverages far more data (hundreds of words, hundreds of users) and presents descriptors to one another using a 2-D map, rather than as a list of presets.

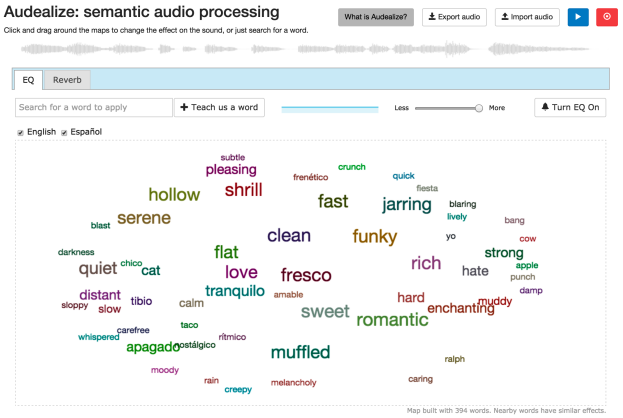


Fig. 3. Audealize, a crowdsourced word-map interface for controlling an equalizer. Above the map are the waveform visualization, seek controls, play/stop, record, import, and export buttons. The map displays learned concepts for equalization. Users can explore the map (shown here with both English and Spanish words), or use the search box.

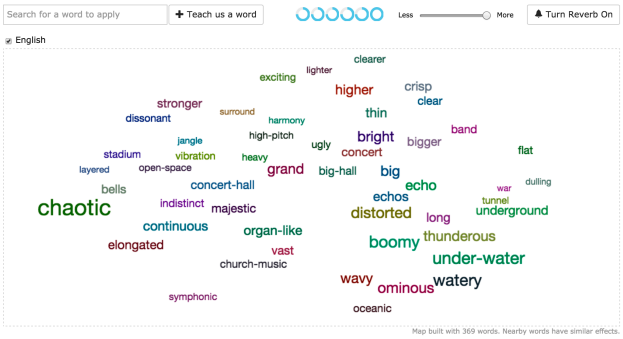


Fig. 4. Audealize, a crowdsourced word-map interface for controlling a reverberator. The map displays learned concepts for reverberation. Users can explore the map, or use the search box.

2 Crowdsourcing Audio Concepts

Since our approach depends on crowdsourced collection of mappings between words and effects control settings. We now provide a brief overview of the SocialReverb [4] and SocialEQ [5] user interactions for data collection. For further detail, please refer to the prior publications.

2.1 SocialEQ

SocialEQ is a web based project for learning a vocabulary of actionable audio equalization descriptors. Since deployment, SocialEQ has learned 792 distinct words in 2327 sessions. Of these, 394 words are in Audealize based on inclusion criteria described in Section 3.5.

In SocialEQ, mappings between words and equalization settings are taught to the system by users. A user of SocialEQ will first provide a freely-chosen word they wish to teach to the system (e.g. “bright”). They are then asked to listen to multiple equalization effects as applied to some reference audio and rate effect in terms of how

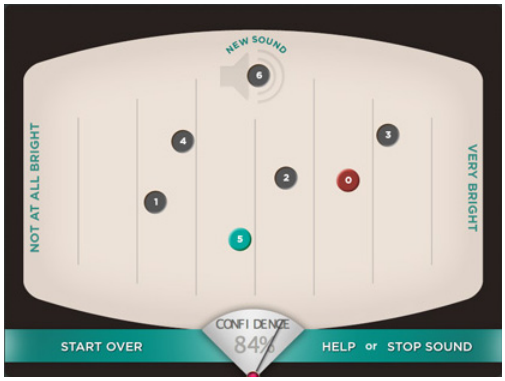


Fig. 5. SocialEQ: Participants are asked to rate equalization examples in terms of how well each matches their goal audio concept.

well it matches their chosen word (e.g. from "not at all bright" to "very bright". These ratings are used to interpolate an actionable equalization definition for the chosen word, personalized to the users taste. When multiple users teach SocialEQ the same word, we can take the average of their equalization settings to find a crowdsourced definition of the word. The public can contribute to SocialEQ at www.socialeq.org, and the interface is shown in Figure 5.

2.2 SocialReverb

SocialReverb is a project to crowdsource a vocabulary of audio descriptors that can be mapped onto concrete actions using a parametric reverberator. We deployed SocialReverb on Mechanical Turk where 513 unique users described 256 instances of reverberation using 2861 unique words, from which the 369 words that showed broadest agreement among participants [4] were selected for use in the current work.

In SocialReverb, mappings between words and reverberation settings are taught via a labeling task. A contributor to SocialReverb is simply presented a reverberation effect and asked to describe it in their own words. They are then presented a list of words other people used to describe the same reverberation effect and asked to check off the ones they agree also describe the effect. This elicitation process results in a crowdsourced labeling of the effect space, giving us many pairings of words and effects. For each effect, a list of words is now available to describe it. To build an actionable reverberation definition for a given word (e.g. "church"), we take all effects that were labeled with the word and take the mean of each reverberation measure (see Section 3.1) to find the most generalizable effect. The public can contribute to SocialReverb at www.socialreverb.org, and the interaction is shown in Figure 6.

3 Audealize

We now describe Audealize, an interface for reverberation and equalization based on natural language adjectives a non-expert would use to describe the audio effects (see Figures 3 and 4). The user modifies a sound recording by clicking on descriptive terms (e.g. "warm", "tinny",

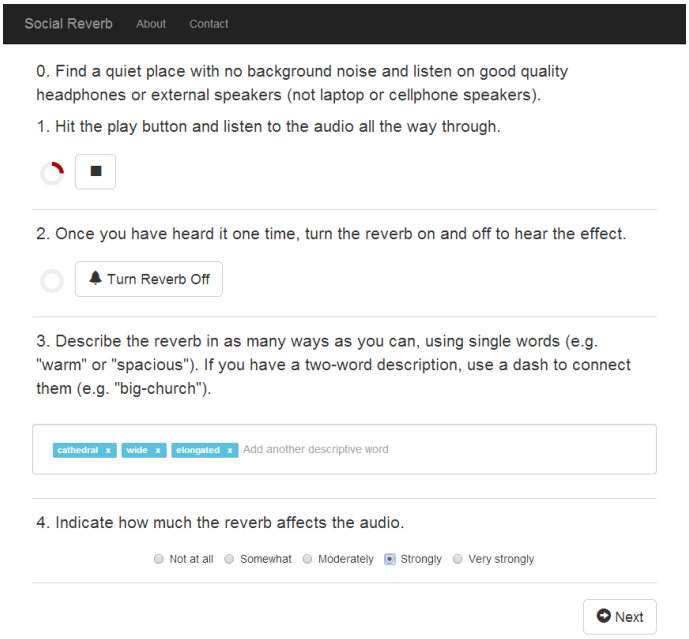


Fig. 6. SocialReverb: Participants are asked to listen to a dry recording, then a recording with an audio effect applied, and then describe it in their own words.

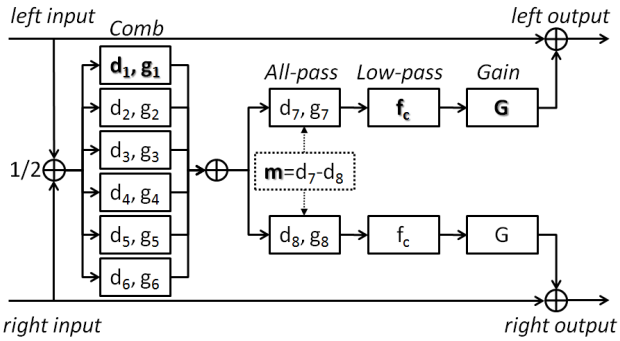


Fig. 7. The digital stereo reverberation unit. Traditional reverberation units have controls that are based directly on gain and delay values within this circuit, rather than on perceptually relevant features.

"chaotic") in a 2-D word-map. When a word is selected, the appropriate audio manipulations are applied to modify the sound to embody the descriptive term.

Audealize integrates our prior work in active and transfer learning of audio concepts ([5], [17], [18]) with crowdsourced vocabularies relating the signal-parameter space to the programmatic space ([4], [5], [19]) to create a cohesive word-map interface to control multiple audio production tools. This bridges the gap between the *programmatic space* and the *signal-parameter space.*

3.1 Underlying audio effects

While a word-map could be a front end to any parametric reverberator or equalizer, we use the same reverberator as used in SocialReverb [4], and a similar equalizer to the one used in SocialEQ[5]. We chose these reverbera-

tor and equalizer implementations since they are comparable to state-of-the-art implementations used in professional digital audio workstations, such as ProTools.

The reverberator is controlled by five parameters: gain, delay, channel delay, frequency cutoff and wet/dry gain. A reverberation effect is described by five measures: reverberation time, echo density, clarity, central time, and spectral centroid. It is shown in Figure 7.

The equalizer is controlled by forty second order filters, each centered around logarithmically spaced frequency bands. At each band, we place a peaking filter at its center frequency. The strength of each filter is controlled by a gain value. Each equalization setting is then described by forty gain values, with each gain value corresponding to a boost (positive gain) or a cut (negative gain) at a particular frequency.

Both the reverberator and the equalizer are implemented entirely within the browser, using the Web Audio API, a graph-based audio processing framework deployed in modern browsers [20].

3.2 Vocabulary

While one can use any vocabulary that maps words onto parameter settings, we seed the word-map with the crowdsourced vocabulary from the target population described in Section 4 drawn from the SocialReverb [4] (369 words) and SocialEQ [5] (394 words) data collections.

Our interface paradigm is language-agnostic. For example, To change the language from English to Spanish [5], one need only substitute in a vocabulary with mappings between effects and words in the new language. When the vocabulary for an additional language is detected, Audealize offers the user the option of selecting it.

3.3 Word-map

We create a 2 dimensional word-map for each production tool: one for reverberation and one for equalization. The map for equalization is shown in Figure 3. The map for reverberation is shown in Figure 4.

Recall that each word in the vocabulary for a production tool has a mapping to the parameters required to elicit that word. We construct the map for each using multi-dimensional scaling [21]. Multi-dimensional scaling is a method for projecting a higher dimensional space into a lower dimensional one, by preserving the distances between elements in the space. For reverberation, we map a 5 dimensional feature space (the five parameters controlling the reverberator) to a 2 dimensional space. For equalization, we map a 40 dimensional space (40 frequency bands) down to a 2 dimensional space. The result is a 2 dimensional map for each effect where closely related effects are placed near one another. The user may click on a word to hear the resulting audio effect. The size of the word on word-map correlates with the consistency of the definition for that word in the crowdsourced data.

Users explore the space of effects by clicking and dragging around the map, using the words as a guide. When a word is selected, the associated effect is applied to the sound. Words near each other have similar effects.

3.4 Search

If the user would rather quickly jump to a concept, such as “boomy”, “underwater” or “shrill”, they can type the word into the search box. If the interface knows the relationship between the descriptor and the audio effect setting, we jump to it immediately. If the interface knows the relationship, but for the other effect (e.g. the user is searching on the reverb map, but the word is found on the equalization map), it suggests that they switch effects.

If the user types in a word not found on either effect map, the interface searches WordNet [22] to find the closest available synonyms on the map and suggests those. For example, if the user types “shrill” into the reverberation search box, the interface responds with “Try sharp, high, high-pitched, or try checking the EQ map”. We also suggest that the user can try teaching the word to Audealize.

3.5 Teaching and updating the interface

If there is no word or synonym on the word-map that adequately describes the user’s concept, the user may teach a new word to Audealize. The system asks the user to rate a series of audio manipulations of a sound in terms of how well they evoke the concept. This training method is based on the one used in [5] and [17]. To speed learning, we use the active and transfer learning techniques developed in [18] and inspired by [23]. This lets Audealize learn the mapping between a word and an equalization setting after only 8 user-rated examples. For reverberation, the user must rate 18 examples. These examples are chosen to cover the relevant space of possible equalization and reverberation effects.

At the end of the teaching process the training data is stored on our server, the system asks how well the learned effect matches the user concept and the word is immediately included in the local map for the user.

Each day, the system automatically pulls all training data from the database, and if a learned word passes inclusion criteria, it is incorporated into the standard map for all users. The inclusion criteria depends on the data source. For SocialReverb, we use the inclusion criteria described in [11]. For SocialEQ, we use the inclusion criteria described in [5]. For data coming from Audealize, we use inclusion criteria incorporating the number of users who have taught Audealize the word (at least 2), how long the average user took to teach Audealize the word (at least 50 seconds), and the number of examples rated by each user during teaching (at least 5). These are proxies for the effort put into teaching Audealize. Only words with high-effort teaching sessions from multiple users are incorporated into the standard map. The size of the word on the map is scaled by the agreement between users.

3.6 Incorporating a signal-parameter interface

Below the word-map interface we provide a button that, if clicked, reveals a traditional interface of signal-

parameter controls (see Figures 11, 12). This lets users explore the space of effects with the map, and then, if they desire, refine the effect using the signal-parameter interface. As a word is selected in the map, the traditional controls are updated to reflect the word's settings. As the effect is manipulated with the signal-parameter interface, the most closely related word in the map is highlighted. This helps the user learn mappings between the signal-parameter space and the descriptive words the general public would use to describe the resulting effect.

Audealize can be accessed at http://audealize.appspot.com. It works entirely inside the browser and is best viewed via recent versions of Google Chrome.

4 Crowdsourced vs. Preset Vocabularies

Our goal is to build tools for a target population that are not expert audio engineers. Our approach is to parameterize the tools in terms of programmatic vocabulary that is easy for them to understand. The most straightforward way of doing this is to learn the vocabulary from the target population. To define a vocabulary of audio terms for laypeople, we used the crowdsourced vocabularies collected in the SocialEQ [5] and SocialReverb [4] projects. Both crowdsource a vocabulary from hundreds of Amazon Mechanical Turk workers. As Mechanical Turk sources workers from the general population, it is a reasonable population for a tool designed for non-experts in audio processing.

While the exact vocabulary may vary if one were to select a population of acoustic musicians, we expect the broad results to be similar, since neither group tends to have expert knowledge nor expert vocabulary for sound engineering. If one were to expect a population to have a largely different vocabulary (e.g. Spanish speaking acoustic musicians vs English speaking electrical engineers), one would crowdsource a new vocabulary as done in SocialEQ [5] and SocialReverb [4].

Existing popular audio production tools attempt to simplify interfaces by providing expert-generated lists of preset effects combinations (preset lists) labeled with natural language terms, such as "clear." These preset lists align words with tool settings and have both the words and actionable definitions (effects parameter settings) for what the words mean. Why, then, not use one of these readily available vocabularies? To establish whether a vocabulary drawn from the preset lists of existing tools would be useful for a given target population of users we need to answer the following question: How large is the shared vocabulary between the preset lists and the target population?

SocialEQ [5] provided a crowdsourced vocabulary of 394 user-generated adjectives for equalization (e.g. "warm"), giving the relationships between each word and the settings for a parametric equalizer. SocialReverb [4] crowdsourced a similar relationship between a vocabulary of 369 user-generated adjectives and the settings for a parametric reverberator.

We collected the preset audio production vocabularies by examining preset lists in popular commercial equalization and reverberation tools. This dataset consists of 168 descriptors for reverberation and 239 descriptors for equalization, and was collected from two popular, state-of-the-art audio production environments - Adobe Audition and Ableton Live. An equalization or reverberation setting is associated with each of these descriptors.

Fig. 8. Overlap between three sets of vocabulary for reverberation - preset vocabulary from Ableton (69 words), Audition (120 words), and the vocabulary used by the target population (369 words).

Figure 8 shows the overlap between the three reverberation vocabularies we collected - words in preset labels from Adobe Audition, words in preset labels from Ableton Live, and words used by our target population. Only 15 words are shared between all three vocabularies: Ableton (69 words), Audition (120 words) and the target population (369 words). Between the two preset lists, only 24 words are shared. If we combine the Ableton and Audition vocabularies and call it the "preset" vocabulary, just 38 words are shared between presets and the target population. This is just 10.2% of the total vocabulary used by the target population.

Figure 9 shows the overlap between the three vocabularies for equalization. Only 5 words are shared between all three vocabularies: Ableton (73 unique words), Audition (197 words) and the target population (394 words). The Ableton and Audition vocabularies share only 31 words. If we combine the Ableton and Audition vocabularies and call it the "preset" vocabulary, only 27 words are shared between the experts and target population, making for just 6.8% of the total target population vocabulary. This indicates a similar problem as with reverberation - the tool builders don't share language with each other, or with our target population.

Most words used by the target population are not found in the preset lists. What about the overlap words shared between Abelton, Audition and laypeople? Do all three share a definition for each word? For reverberation, we take each impulse response associated with a descriptor in preset vocabulary and calculate a descriptor definition using the re-

Fig. 9. Overlap between three sets of vocabulary for equalization - preset vocabulary from Ableton (73 unique words), Audition (197 words) and the target population (394 words)

verberation time, echo density, clarity, central time, and spectral centroid as in [4]. We then find the distance between the preset definitions and the crowdsourced definition for a descriptor. The smallest distance between definitions for a descriptor found in both vocabularies was for the descriptor "rich". This distance corresponds to .875 seconds difference in reverberation time, a difference in the cutoff frequency of over half a musical octave (451 Hz to 277 Hz), and a 14 decibel jump in clarity. Each of these differences means a drastic difference in how the audio effect is perceived by listeners. This indicates that even when the tool-designers and the target users (laypeople) share a word, it is unlikely that the two populations agree on what the word means.

Results are similar for equalization. For each shared equalization descriptor, we compare each preset equalization curve, with the crowdsourced equalization curve using Pearson correlation. We find the definitions are poorly correlated, with the best correlation found to be .438 ("dark"), and the worst to be $-.164$ ("funky"). What is more, the definition used in Ableton Live frequently did not agree with that from Adobe Audition. An example of this for equalization is shown in Figure 10. Here one preset (Ableton's) shows a rough agreement with the crowd, while the other (Audition) has a very different equalization curve.

The weak overlap between crowdsourced and preset vocabularies speaks to a disconnect between tool builders and lay populations when describing audio effects. Therefore, an interface using preset vocabulary would be ill-suited for use by our target population (and, likely other non-experts). For this reason, we use a vocabulary grounded in descriptors crowdsourced from the target population, rather than one curated by the professional toolbuilders. To build a vocabulary for a different target population (e.g. professional recording engineers, orchestral musicians), one would run a data collection on that target population and build the interface from the resulting data.

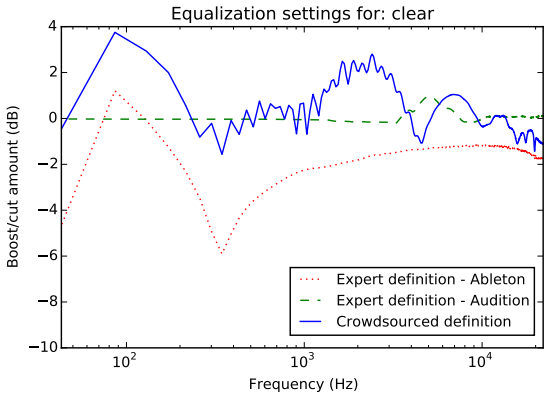


Fig. 10. Comparing the concept 'clear' in presets in professional audio production environments and as learned from the target population.

5 User study

We have established why we use a vocabulary learned from the target population. Audealize leverages word learning and crowdsourcing approaches to collecting this vocabulary that were validated in prior work. Therefore, we focus on the efficacy of the crowdsourced word-map interface created using this vocabulary compared to signal-parameter interfaces. The study seeks to answer the following questions:

Q1. Do participants prefer a crowdsourced word-map to a signal-parameter interface for programmatic audio production?
Q2. Do participants prefer a crowdsourced word-map to a signal-parameter interface for non-programmatic audio production?
Q3. Do participants show better absolute performance using a word-map for non-programmatic audio production?
Q4. Does the effectiveness of the word-map degrade gracefully as the task increases in difficulty?

5.1 Experimental Design Overview

We designed a study that balances a number of factors. There are two production tools: an equalizer and a reverberator. There are two interfaces for each tool: a crowdsourced word-map (Audealize) and a signal-parameter interface. There are two tasks - *match-word* (which simulates a programmatic audio production task) and *match-effect* (which simulates a non-programmatic production task). There are two audio files to modify. Each task has three difficulty levels. There are three variants at each of the three difficulty levels. Each participant always did one task from each of the three difficulty levels. Therefore there are $2 * 2 * 2 * 2 * 3^3 = 432$ unique conditions in the experiment. We now describe the flow of a single participant session. Detailed descriptions of tasks, audio files, etc. follow.

Each person participates in one session dedicated to either the match-word task or the match-effect task. Each session (whether for the match-word or match-effect task) has the same outline:

1. *Selection of Conditions*: The system randomly selects which audio file to use. A set of 3 (one hard, one intermediate, one easy) effects (for match effect) or words (for match-word) is selected. These are held constant for both interfaces. The ordering of control and test interfaces is randomly selected. The production tool to use (reverberation or equalization) is chosen by the system.
2. *Training on Both Interfaces*: The participant is presented with the chosen audio file. They are given two minutes to play with the audio using the production tool via either the signal-parameter interface or the word-map interface. They then have two minutes to use the other interface. Presentation order between interfaces is randomized.
3. *Interface A*: Either the control or the test interface is selected randomly. The participant performs the test on the hard, then the intermediate, and finally the easy condition. After each condition, the participant gives feedback on the task (see below)
4. *Interface A feedback*: The participant gives overall feedback about Interface A (see below).
5. *Interface B*: The participant performs the test with the other interface. The effects, audio, order of presentation and feedback prompts are identical to those for Interface A.
6. *Interface B feedback*: The participant gives overall feedback about Interface B.
7. *Exit comments:* at the end of the survey, we ask the participant to provide any comments on either interface that they feel it would be helpful to share.

We selected the hard, intermediate, easy order of tasks to provide the most extreme range of difficulty for the test. Placing the hard task first ensures minimal training and the most difficult scenario. Placing the easy task last ensures maximal learning as well as the easiest scenario. Since a pilot study showed a single user session (three tasks on two interfaces for a given audio production tool and given sound file) takes on the order of 10 minutes, we felt that fatigue was unlikely to be a strong factor.

After each task in a session, the user is presented with the following statements:

- I achieved the goal.
- I was satisfied with my experience using this interface.
- I was able to find relevant audio effects easily.

For each statement the participant is asked to express a level of agreement using a continuous slider ranging from 0 (disagree) to 1 (strongly agree).

Once all three matching tasks (hard, intermediate, easy) are complete, the user is presented two final statements and asked to express their level of agreement:

- I enjoyed using this interface.
- I understood how to use the interface to achieve a specific goal.

Finally, participants were provided a free-form comment box where they could leave feedback.

5.2 Match-word task

In the match-word task, we present an audio recording with no audio effects applied to it, along with a descriptive word (e.g. “underwater”). The participant is asked to imagine what audio manipulation embodies the word and use an audio production tool (e.g. reverberation or equalization) to alter the recording to make it embody the descriptive word. The interface for the tool is either the test (crowdsourced word-map) or control (signal-parameter) interface. This simulates a programmatic audio production task, which Audealize was designed for.

This task has three levels of difficulty. The hardest is where the word is not present on the map. Here, participants must find a synonym in the map to get at the same effect. To construct this case, we take a word in the map, and delete it from the interface. In the intermediate and easy levels, we ask the user to match words that are in the map. The intermediate level draws from a set of medium confidence words, and the easy level draws from a set of high confidence words. Each difficulty level is associated with a set of three words, from which one is chosen for the participant. For reverberation, the three sets are: easy - *chaotic, underwater, boomy*, intermediate - *underground, bells, crisp*, and hard (not shown on map) - *crashing, oceanic, tunnel*. For equalization, the three sets are: easy - *tinny, hollow, deep*, intermediate - *harsh, clear, muffled*, and hard (not shown on map) - *shrill, clean, mellow*.

5.3 Match-effect task

In the match-effect test we present the user with two variants of the same underlying audio file: the original file, and a version altered using a production tool (e.g. a reverberator). The user is then presented the same effect tool used to produce the altered file. The interface for the tool is either the test (crowdsourced word map) or control (signal-parameter) interface. They are asked to use the tool to modify the original file to match the altered file. This simulates a common audio production task - replicate an effect heard in an existing recording.

Matching an effect is the production task that signal-parameter interfaces are designed for. As such, we would not expect Audealize to out-perform traditional interfaces on any version of the task. We can, however, choose to make the target sound be one modified by an effect corresponding to a word in the map. This lets us create three difficulty levels: hard (there is no corresponding word on the map), medium (there is a corresponding word on the map, but it is not prominent on the map), and easy (the corresponding word is prominent).

5.4 The Control and the Test Interfaces

In the control condition, the participant is provided a signal-parameter interface with 5 parameters (for reverberation) or a signal-parameter interface with 40 parameters (for equalization). These interfaces directly control the un-

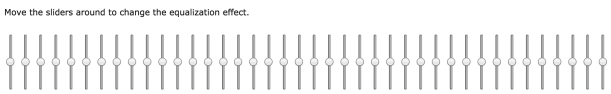


Fig. 11. The signal-parameter interface for an equalizer for the user study. The effectiveness of this interface for the matching tests was compared to the word-map interface, shown in the upper half of Figure 3.

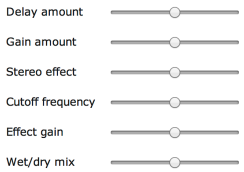


Fig. 12. The signal-parameter interface for a reverberator for the user study. The effectiveness of this interface for the matching tests was compared to the word-map interface, shown in the upper half of Figure 4.

derlying parameters of the effects tools. The control reverberation interface is shown in Figure 12. The control equalization interface is shown in Figure 11.

While more advanced interfaces for reverberation and equalization exist, such as ones that allow you to draw the equalization curve or parametric equalizers that give knobs to control center frequencies and Q, these interfaces still force the user to navigate the signal-parameter space rather than the programmatic space. Graphic equalizers and parametric equalizers both force the user to think in terms of frequency bands and manipulating their levels. This is also true for interfaces that let you draw the EQ curve. We believe this makes them all fundamentally the same in the underlying mental model that must be created to use the interface effectively. A graphic equalizer is one well-known representative interface of this interaction model and so is the one used to compare to in the study.

In the test condition, the participant is shown a (slightly) simplified version of the Audealize interface. The crowdsourced word-map and search box are available, along with the effect on/off and gain. The "show traditional interface" and "teach us a word" parts of the interface are removed.

5.5 The Audio Data

We used two audio files for all tasks in this study: a 10 second recording of an electric guitar (solo) and a 14 second recording of an acoustic drum kit (solo), both recorded at a bit depth of 16 and a sample rate of 44100 Hz. These two audio recordings were also used in the SocialReverb data collection [4]. The recordings were encoded as .mp3 [24] at a bitrate of 320kbps. Painter and Spanias [25] showed that this encoding is perceptually indistinguishable from uncompressed audio.

5.6 Participant recruiting and inclusion criteria

We recruited participants for our user study from Amazon Mechanical Turk [26]. Mechanical Turk provides measures of worker reliability that we used to pre-screen participants. We only allowed workers with a 97 percent positive review rating and who had performed at least 1000 tasks on Mechanical Turk. A study of the demographics of Amazon Mechanical Turk workers [27] indicates that the majority of workers are from the US and have a roughly equal gender balance.

Our second inclusion criterion was passing a hearing test, based on one used in a prior web-based study [4]. The participant is presented two sequences of tones, each of which has between 2 and 8 tones in ranging in pitch from 50 Hz to 7000 Hz, and must report how many tones are in the sequence. For each sequence, they must report the correct number of tones.

In total, there were 432 participant sessions for the user study - 108 each for match-effect equalization, match-effect reverberation, match-word equalization and match-word reverberation. Each participant passed the hearing test and was also confirmed by Amazon as having a good work record.

5.7 Performance Measures

For statistical hypothesis testing, we use the paired sample Wilcoxon signed-rank test, with samples being paired within a single participant session (a crowdsourced word-map interface versus a signal-parameter interface for the same word or effect and the same audio example.

We analyze the results from our user study in two ways: self-reported performance and absolute performance. For self-reported performance, we look at mean user responses to the prompts described in Section 5.1. User responses range from 0 (strongly disagree) to 1 (strongly agree). This is applicable for both the match-word and match-effect tests on both reverberation and equalization tools.

The absolute performance compares the user-generated audio effect to the target audio effect. As such, it only applies to the match-effect test. For each effect (reverberation and equalization) we must define a measure that expresses a the difference between the goal effect, g, and the user-provided effect, u.

We characterize reverberation as was done in previous studies [4] [28]. We measure five perceptually-relevant features from the impulse response for a reverberation effect - reverberation time (RT60), echo density, clarity, central time, and spectral centroid. Therefore, each reverberation effect is characterized by a five-dimensional vector.

For reverberation, we use a normalized Euclidean distance to measure the absolute quality $Q(g,u)$ of a user attempt to match a goal effect. After calculating distance $d(g,u)$ between each pair of goal and user effects in a test, we scale distances. We then subtract from the maximum distance to get a performance measure ranging from 0 (worst) to 1 (best).

For the absolute measure on the match-effect task for equalization, we use Pearson correlation, rather than Euclidean distance, since it captures relative trends, rather than absolute distance, which is important when comparing equalization curves. With this distance measure, getting the exact boost or cut of each frequency measure is not re-

quired for good performance. Instead, the participant just has to match the relative shape of the equalization curve for good performance. Each equalization curve consists of a 40 dimensional vector, where each dimension indicates the boost or cut of a particular frequency. This is distinct from the reverberation case, where absolute position in the feature space is the best measure. Note, this measure ranges from -1 (inverse correlation) to 1 (perfect correlation).

5.8 Results

5.8.1 Self-reported results

Table 1 shows the mean user responses to the prompts for both the match-effect and match-word tasks on both equalization and reverberation. In every comparison, the mean response for the word-map is on the left and the mean response for the signal-parameter interface is on the right (e.g. [word-map vs. signal-parameter]).

Recall that the match-word task is designed to model a programmatic audio production task, where the goal is to find an effect that evokes a word. The mean user responses are higher for the word-map on all prompts and both audio production tools. In every case, the difference is statistically significant at the $p < 0.05$ level. This indicates that the word-map is preferred over signal-parameter interfaces for programmatic audio production, regardless of whether the tool is an equalizer or a reverberator. This answers **Q1** in the affirmative. A word-map is preferred for programmatic audio production by this population, regardless of underlying tool.

The match-effect task is one where the goal is to take an unmodified recording and change it to match a recording that has been modified with an audio effect. We expected signal-parameter interfaces to be more well-suited for this task than a word-map.

For reverberation on the match-effect task, the average user response to all questions was higher for the word-map than for the signal-parameter interfaces. This said, the difference was only statistically significant for the question "I was satisfied with my experience using this interface." For equalization on the match-effect task, we find statistically significant difference in favor of the word-map at the $p < .05$ level for all questions. This is a surprising result, since we expected the signal-parameter interfaces to perform significantly better on the match-effect task. This indicates that a crowdsourced word-map may be better suited for equalization by non-experts than signal-parameter interfaces, when the user is a non-expert.

Thus, for non-programmatic production, users find the word-map to be an effective alternative to the signal-parameter interface (**Q2**), and it has been shown to beat the signal-parameter interface convincingly on the equalization task. Here, the production tool matters. One possible explanation for the difference is that the number of frequency bands on equalization is 40, while the reverberator had only 5 controls. Perhaps the advantage of a word-map increases as the number of controls to be manipulated increases. A future study will be required to determine this.

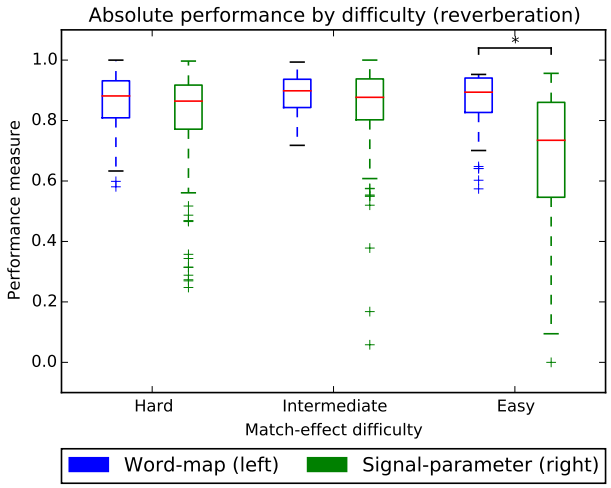


Fig. 13. Match-effect test: absolute performance for reverberation broken down by task. The * indicates statistical significance between distributions at a p-value $< .01$. We find significant difference in favor of the word-map interface in the easy task. $N = 108$ for each box plot.

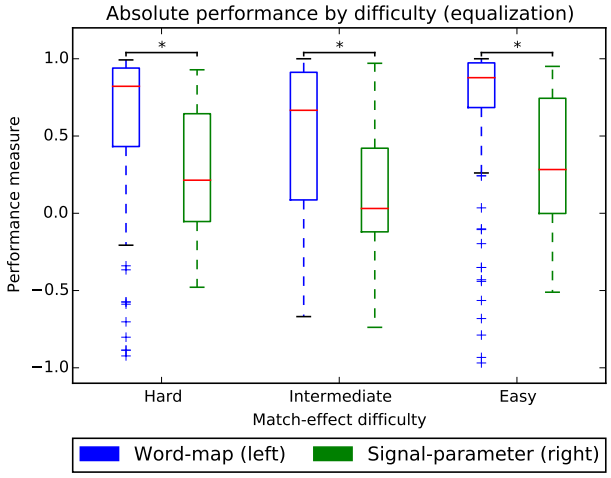


Fig. 14. Match-effect test: absolute performance for equalization broken down by task. The * indicates statistical significance between distributions at a p-value $< .01$. $N = 108$ for each box plot. We find statistically significant differences across all three tasks.

5.8.2 Participant Comments

In the freeform response box provided to study participants, many people wrote comments that indicate they found the word-map approach, which uses lay vocabulary to control audio effects, to be a compelling alternative to the traditional signal-parameter approaches. Representative quotes include:

"I *love* the word interface! I am a musician and also a writer who has written professionally about music, and the word interface matches the way my brain works, perfectly. Where/when can I get this?"

"I hate the traditional EQ interface and always have difficulty getting what I want, but this one absolutely makes sense for me. So cool!"

Prompt	Mean response (reverberation, match-word)	Mean response (reverberation, match-effect)	Mean response (equalization, match-word)	Mean response (equalization, match-effect)
I achieved the goal.	**.833** vs .712	.707 vs .668	**.867** vs .648	**.806** vs .574
I was satisfied with my experience using this interface.	**.768** vs .679	**.653** vs .596	**.842** vs. .531	**.812** vs .464
I was able to find relevant audio effects easily.	**.749** vs .628	.557 vs .543	**.842** vs .499	**.762** vs .386
I enjoyed using this interface.	**.782** vs .688	.664 vs .573	**.876** vs .495	**.862** vs .420
I understood how to use this interface to achieve a specific goal.	**.856** vs .735	.768 vs .715	**.888** vs .574	**.886** vs .639

Table 1. Mean user response to select prompts for match-word and match-effect. Each column is formatted as '[word-map] vs [signal-parameter]'. Responses could range from 0 (strongly disagree) to 1 (strongly agree). Boldface indicates statistical significance between the user responses for the prompt at a p-value < 0.05. $N = 324$ per condition for the upper three prompts, $N = 108$ per condition for the lower two.

"I felt like I just didn't even know where to start with Interface B [signal-parameter], and that I could play around minutely with all of the sliders for hours and still not be able to match the effect. It frustrated me very quickly."

"Wow, I can't even begin to explain how much I liked Interface A [word-map] better than Interface B [signal-parameter]. I always dreamed of having a big old equalizer in my stereo system after I graduated from college, now I can only imagine what a nightmare it is to adjust it every time you want to achieve some type of sound. Interface A [word-map] makes intuitive and visual sense. Interface B [signal-parameter] made me want to tear my hair out."

"I know the second interface [signal-parameter] allows fine tuning and is probably better for someone experienced with this type of activity, but since I'm not, I found the first interface [word-map] much easier to work with and more intuitive to use."

The comments indicate that there is a population that would like interfaces such as the word-map as an alternative to the existing signal-parameter interfaces. Comments also indicate that a mix of the two interfaces is desired, with the word-map used for high-level control, and the signal-parameter for fine-tuning. This feature was not in the test interface for the experiments, but is in the deployed version of Audealize.

5.8.3 Absolute Results

For the match-effect task, we can augment self-reported results with absolute measures. Figure 13 shows how closely each user matched the target reverberation effect using the word-map interface versus the same person using the signal-parameter interface.

For reverberation, we find in the easy task, where the effect matches one associated with a high-confidence word on the map, performance skews significantly higher for our interface than the signal-parameter interface. In the hard and intermediate cases, the performance of both systems were not statistically distinguishable (**Q3**). Figure 14 shows absolute performance for equalization. For equalization, we find statistically significant difference (at the $p < .01$ level) performance on all three tasks for in favor of the word-map.

Those using the word-map interface also show no degradation in performance as the task difficulty increases (**Q4**). This holds true for both reverberation and equalization. Since difficulty levels are equal in all cases for the signal-parameter interface, we do not expect and do not see meaningful differences in performance for this interface. The signal-parameter interface, however, does show worse performance on the easy task. We did not expect to see this result. Recall that our design ordered the tasks hard, intermediate, easy. The intent of this was to magnify the differences between difficulty levels by putting the easy case last, increasing the learning effect. We did not expect a fatigue effect, given that the active part of a participant session (doing the three tasks per interface for both interfaces) takes an average of about ten minutes.

We speculate that there may be a fatigue effect in matching reverberation effects for a novice that becomes evident for the signal-parameter interface sooner than it does for the word-map. This fatigue effect is not the result of ordering the signal-parameter interface second. Interface order was counterbalanced, with half the trials putting the signal-parameter interface first. These results indicate that a more challenging task will have to be designed to study the effects of mismatch between the words in the interface and the goal effect. A further experiment will be required to test for interface fatigue.

For equalization, the median time to completion for the signal-parameter interface was 77.91 seconds, and the median time to completion for the word-map interface was 56.28 seconds. For reverberation, the median time to completion for the signal-parameter interface was 71.35 seconds, and the median time to completion for the word-map interface was 61.43 seconds.

6 Conclusions and Future Work

Participants prefer the word-map interface on the match-word task. This was expected, as it is what the interface is designed for: programmatic audio production. The relative effectiveness of the word-map interface for the match-effect task, which it was not explicitly designed for, is surprising. One would expect the fine control afforded by the signal-parameter interface would let the user match the effect more closely than would be likely with the word-map. It may be that for novices, the *signal-parameter space* is harder to navigate than the *programmatic space*, eliminating the advantage of fine-grain control.

This indicates that crowdsourced language, in concert with a meaningful word-map visualization, is an effective interaction paradigm for novice users of audio production tools, for both programmatic and non-programmatic goals.

Audealize is an example of a more general design approach. The crowdsourcing methodologies and interface paradigms used to create the 2D word-map interface of Audealize can be generalized to almost any audio effect, such as compression, chorus, tremolo, etc.

The results from our user study show that word-map interfaces are both preferred by and effective for non-experts. The ideas presented here show a clear path to crafting accessible and effective interfaces in other domains.

Another promising angle to explore is to collect vocabularies from a variety of target populations, including different kinds of musicians (hip-hop, classical, etc), and different linguistic groups (Spanish, Korean, Chinese) By collecting these words, we would increase the accessibility of the interface and find interesting relationships between how audio effects are described in different languages. We are interested in how effectively Audealize crosses linguistic and cultural boundaries. We are also interested in how vocabulary overlaps between audio effects.

The automatically generated, crowdsourced word-map interface embodied by Audealize is a useful alternative interface approach for audio production that may also apply to other domains (e.g. image manipulation) and points the way for rethinking the relationship between interface controls and the underlying tools used for media production.

7 Acknowledgements

Thanks to Bongjun Kim, Mark Cartwright, Olivia Morales, and Zach Puller for helping with interface development, user study design, and overall feedback on Audealize. This work was funded by NSF Grants 1116384 and 1420971.

References

[1] *The Absolute Sound*. Number v. 18, nos. 87-90, p. 113. Absolute Sound, Limited, 1993.

[2] Jon Burton. Ear machine iq: Intelligent equaliser plug-in. `http://www.soundonsound.com/sos/jun11/articles/em-iq.htm`. Accessed: 2015-04-12.

[3] Robin Bigwood. Audio ease altiverb. *Sound on Sound*, May 2002. URL `http://www.soundonsound.com/sos/may02/articles/altiverb.asp`.

[4] Prem Seetharaman and Bryan Pardo. Crowdsourcing a reverberation descriptor map. In *Proceedings of the ACM International Conference on Multimedia*, pages 587–596. ACM, 2014. doi: 10.1145/2647868.2654908.

[5] Mark Cartwright and Bryan Pardo. Social-EQ: Crowdsourcing an equalization descriptor map. In *14th International Society for Music Information Retrieval*, 2013.

[6] Mark Cartwright, Bryan Pardo, and Josh Reiss. Mixploration: Rethinking the audio mixer interface. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 365–370. ACM, 2014. doi: 10.1145/2557500.2557530.

[7] Ohad Fried, Zeyu Jin, Reid Oda, and Adam Finkelstein. Audioquilt: 2d arrangements of audio samples using metric learning and kernelized sorting. In *NIME*, pages 281–286, 2014.

[8] Matt Hoffman and Perry R Cook. Feature-based synthesis: Mapping acoustic and perceptual features onto synthesis parameters. In *Proceedings of the 2006 International Computer Music Conference (ICMC), New Orleans*, volume 33. Citeseer, 2006.

[9] AT Sabin and B Pardo. 2DEQ: an intuitive audio equalizer. *Proceedings of the 7th ACM conference on Creativity and Cognition*, 2009. doi: 10.1145/1640233.1640339.

[10] S Mecklenburg and J Loviscach. subjEQt: Controlling an equalizer through subjective terms. *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, 2006. doi: 10.1145/1125451.1125661.

[11] Prem Seetharaman and Bryan Pardo. Reverbalize: a crowdsourced reverberation controller. In *Proceedings of the ACM International Conference on Multimedia*, pages 739–740. ACM, 2014. doi: 10.1145/2647868.2654876.

[12] Brian L Schmidt. A natural language system for music. *Computer music journal*, pages 25–34, 1987. doi: 10.2307/3680317.

[13] Josh Mycroft and Justin Paterson. Activity flow in music equalization: The cognitive and creative implications of interface design. In *Audio Engineering Society Convention 130*. Audio Engineering Society, 2011.

[14] Ronald Mo, Bin Wu, and Andrew Horner. The effects of reverberation on the emotional characteristics of musical instruments. *Journal of the Audio Engineering Society*, 63(12):966–979, 2016. doi: 10.17743/jaes.2015.0082.

[15] Cheng-Zhi Anna Huang, David Duvenaud, Kenneth C Arnold, Brenton Partridge, Josiah W Oberholtzer, and Krzysztof Z Gajos. Active learning of intuitive control knobs for synthesizers using gaussian processes. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 115–

124. ACM, 2014. doi: 10.1145/2557500.2557544.

[16] Ryan Stables, Sean Enderby, Brecht De Man, Gyorgy Fazekas, and Joshua Reiss. Safe: A system for the extraction and retrieval of semantic audio descriptors. 2014.

[17] AT Sabin, Zafar Rafii, and Bryan Pardo. Weighted-function-based rapid mapping of descriptors to audio processing parameters. *Journal of the Audio Engineering Society*, pages 419–430, 2011.

[18] Bongjun Kim and Bryan Pardo. Speeding learning of personalized audio equalization. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 495–499. IEEE, 2014. doi: 10.1109/ICMLA.2014.86.

[19] Zafar Rafii and Bryan Pardo. Learning to control a reverberator using subjective perceptual descriptors. In *ISMIR*, pages 285–290, 2009.

[20] Chris Rogers. Web audio api. *Draft [online specification], Version*, 1, 2012.

[21] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005. doi: 10.1007/978-1-4757-2711-1.

[22] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. doi: 10.1145/219717.219748.

[23] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511763113.

[24] LAME. Lame mp3 encoder, 1998-2014. URL `http://lame.sourceforge.net/`.

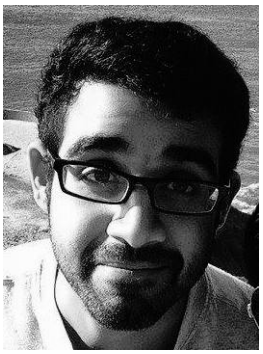
[25] Ted Painter and Andreas Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000. doi: 10.1109/5.842996.

[26] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[27] Panagiotis G Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010. doi: 10.1145/1869086.1869094.

[28] Zafar Rafii and Bryan Pardo. A Digital Reverberator Controlled through Measures of the Reverberation. *Northwestern University, Technical Report*, 2009.

THE AUTHORS



Prem Seetharaman

Bryan Pardo

Prem Seetharaman received a B.S. degree in computer science with a second major in music composition from Northwestern University in 2013. He works on problems in creativity support tools, audio source separation, and machine learning. In addition to research, he is an active composer and musician in the Chicago area.

●

Bryan Pardo received an M. Mus. degree in jazz studies in 2001 and a Ph.D. degree in computer science in 2005, both from the University of Michigan, Ann Arbor. He is an associate professor in the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL. While finishing his doctorate he taught in the Music Department of Madonna University.

Dr. Pardo has authored over 80 peer-reviewed publications and he is an associate editor for IEEE Transactions on Audio Speech and Language Processing. When he is not programming, writing or teaching, he performs throughout the United States on saxophone and clarinet at venues such as Albion College, Chicago Cultural Center, Detroit Concert of Colors, Bloomington Indianas Lotus Festival, and Tucsons Rialto Theatre.