arts

MDPI

Article

Learning to Build Natural Audio Production Interfaces

Bryan Pardo [1,*], Mark Cartwright [2], Prem Seetharaman [1] and Bongjun Kim [1]

[1] Department of Computer Science, McCormick School of Engineering, Northwestern University, Evanston, IL 60208, USA

[2] Department of Music and Performing Arts Professions, Steinhardt School of Culture, Education, and Human Development, New York University, New York, NY 10003, USA

* Correspondence: pardo@northwestern.edu

Received: 11 July 2019; Accepted: 20 August 2019; Published: 29 August 2019

Abstract: Improving audio production tools provides a great opportunity for meaningful enhancement of creative activities due to the disconnect between existing tools and the conceptual frameworks within which many people work. In our work, we focus on bridging the gap between the intentions of both amateur and professional musicians and the audio manipulation tools available through software. Rather than force nonintuitive interactions, or remove control altogether, we reframe the controls to work within the interaction paradigms identified by research done on how audio engineers and musicians communicate auditory concepts to each other: evaluative feedback, natural language, vocal imitation, and exploration. In this article, we provide an overview of our research on building audio production tools, such as mixers and equalizers, to support these kinds of interactions. We describe the learning algorithms, design approaches, and software that support these interaction paradigms in the context of music and audio production. We also discuss the strengths and weaknesses of the interaction approach we describe in comparison with existing control paradigms.

Keywords: music; audio; creativity support; machine learning; human computer interaction

1. Introduction

In recent years, the roles of producer, engineer, composer, and performer have merged for many forms of music (Moorefield 2010). At the same time, software developers have created many tools to enhance and facilitate audio creation as computers have increased in performance and decreased in price. These trends have enabled an increasingly broad range of people, professional and amateur, to use audio production software tools to create music, create sound art, and enhance recordings. Given the broad range of people now using audio production tools, improving these tools provides a great opportunity for meaningful enhancement of creative activities.

When dealing with audio production tools, there are several conceptual spaces the user must be able to negotiate to achieve a goal: the parameter space, the perceptual space, and the semantic space. The parameter space is the space of low-level technical parameters of audio production tools. On a reverberation tool, such a control might be a knob labeled "diffusion". When we hear the audio output of an audio production tool, it is projected to a perceptual space (e.g., we notice longer echos when reverb time increases). Lastly, what we perceive also relates to a semantic space which is how we describe the world (it sounds like a "big cave"). Typically, the goals of a musician or sound artist lie in the perceptual space ("It should sound similar to this <play sound>") or semantic space (e.g., "I want it to sound like they were playing in a small closet and then emerged into a church"), rather than the parameter space (increase the gain at 400 Hz by 12 decibels).

To communicate with current audio production tools, one must understand the parameter space and how it relates to the higher level perceptual and semantic spaces. This mapping is often highly nonlinear and dependent on multiple settings. This can make it difficult to learn mappings between

desired effects ("sound like you're in a cave") and the low level parameters (a knob on a reverberator labeled "diffusion").

Much of the way we interact with today's audio production tools still relies on conventions established in the 1970s for hardware tools used by dedicated audio engineers. Users communicate their audio concepts to complex software tools using (virtual) knobs and sliders that control low-level technical parameters whose names can be difficult to interpret for those not trained in audio engineering and difficult to know how to use when seeking a desired effect.

For example, an acoustic musician may not know what "Q" is, yet a "Q" knob is a common control on a parametric equalizer that determines the frequency bandwidth of a filter (see Figure 1). Once Q has been explained, it still may not be obvious how to use a Q knob, in combination with other parametric equalizer controls to realize a perceptual goal. For example, how would one make the sound "bright" using the parametric equalizer in the figure?[1]. This is a simple example that, while easy for a knowledgeable engineer, can be frustrating for one without the appropriate knowledge and experience.

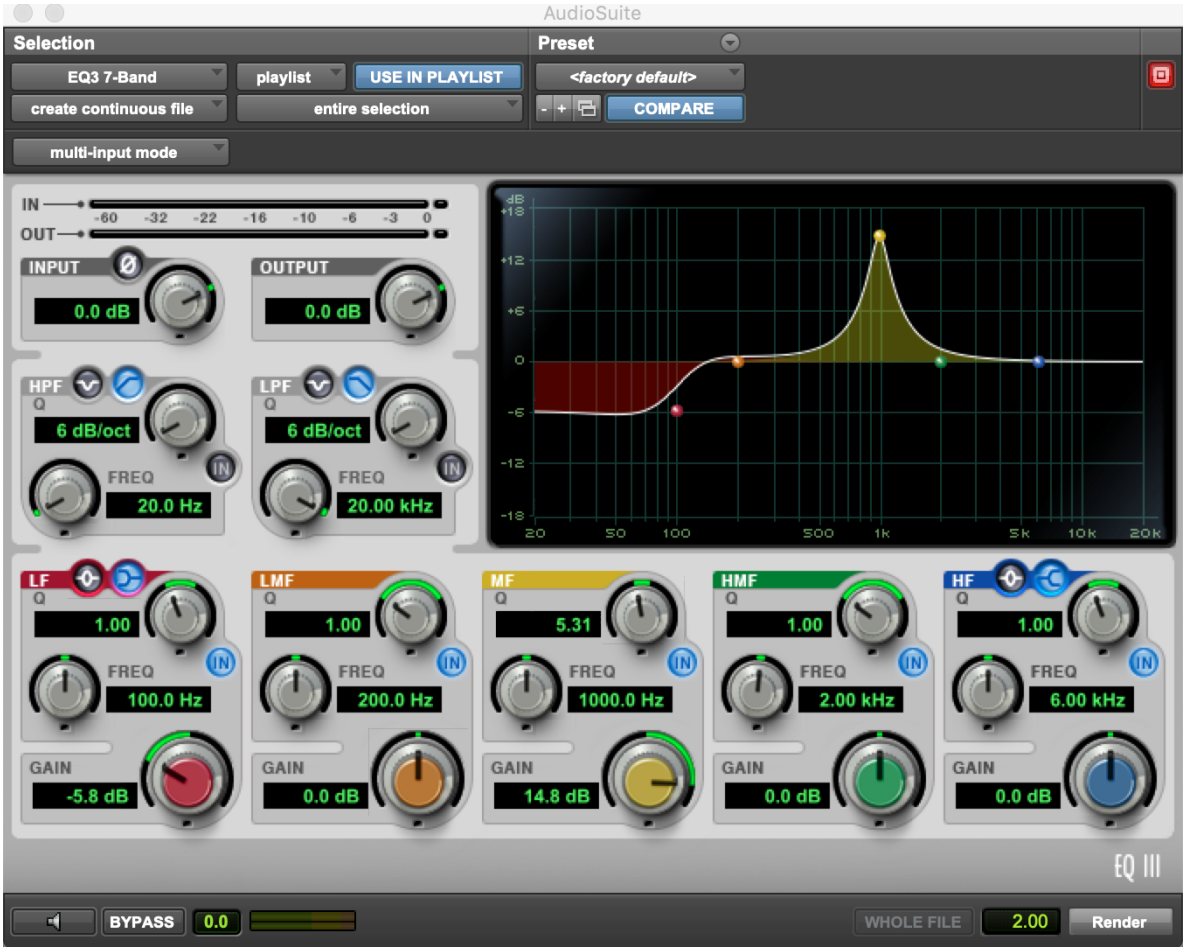

Figure 1. The parametric equalizer included with ProTools Ultimate 2019.6.0. How would you make the sound "bright" using this interface, if you did not already know the mapping between the word and how to change the frequencies? Screen capture created by the authors.

It is often not obvious to a non-engineer (e.g., an acoustic musician) how to manipulate the knobs and sliders to achieve desired effects, and many musicians do not have the time or desire to master

[1] To make a recorded piano sound more "bright" using a parametric equalizer, one might select a medium-high "center frequency" somewhere between 2000 and 4000 Hz, set the "Q" to be very wide and apply a positive "gain".

all the controls available in modern digital audio workstations. This frustrates users and limits their creative output. The following quote from an anonymous Reddit user is exemplary of this situation.

> I have been playing guitar 30 years. I bought the recording interface, software, etc. 6 months ago. As I am 48 and work as a carpenter, I am just too damn tired all the time to learn this stuff. There is so much to learn at the same time, I don't know all the terminology. I have given up for now. Sad, because I have lots of ideas.[2]

Even professional musicians express frustration with this paradigm, embodied by many existing audio production tools (e.g., Abelton Live and Avid's ProTools). Dee Dee of the band Dum Dum Girls has this to say on the subject "I open programs like Ableton and sort of stare mouth agape at the screen" (Tavana 2015). Figure 2 shows a screenshot of Ableton Live 10, the current version as of this writing.

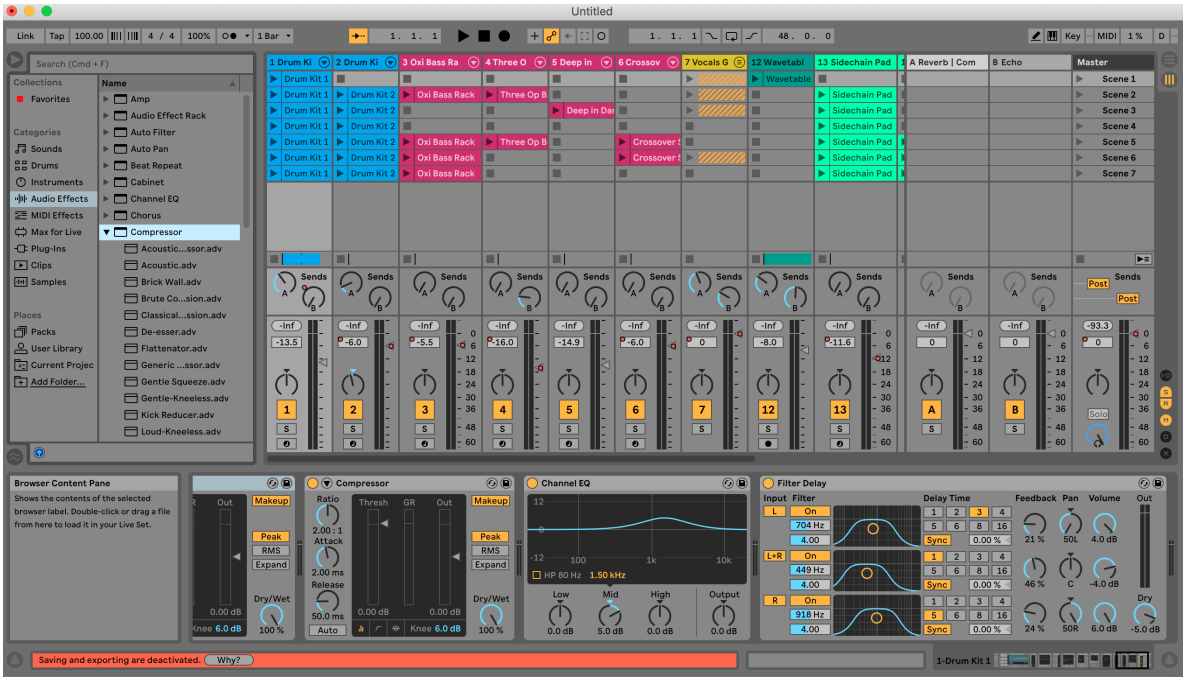

Figure 2. Ableton Live 10, the current version as of the writing of this article. Screen capture created by the authors.

Tool builders are aware of this situation. Some have addressed the issue of difficult-to-use interfaces by removing controls to simplify the interaction (e.g., Apple's GarageBand). Removing needless complexity in an interface can be helpful and empowering. GarageBand, in particular, has brought the tools of digital music making to a broad swath of people who may not have otherwise delved into this art form.

Too much automation, however, can make it harder to innovate, and artists have expressed dissatisfaction with the approach taken by GarageBand. Brooklyn rapper and producer Prince Harvey states "You feel like you're being told what to do now" (Tavana 2015). This sentiment is echoed by Dupuis of the band Speedy Ortiz, who states "I feel like the new GarageBand is devaluing my intelligence in a way" (Tavana 2015).

Removing user controls and automating processes is, perhaps, most fully realized in products like those of LANDR. LANDR provides a fully automated music mastering tool that applies equalization, compression and other effects with no input from the artist. This can be a great help if the artist doesn't

---

[2] https://www.reddit.com/r/WeAreTheMusicMakers/comments/21u3ia/if_you_want_to_learn_how_to_produce_music_you_can/.

want to address some aspect of the creative process (e.g., mastering), but the end game of this approach if applied to all aspects of creating the work, is to remove the artist from the process entirely.

Our design philosophy is fundamentally different from the ones described above. In our work, we use metaphors and techniques familiar to musicians to produce customizable environments for music creation, with a focus on bridging the gap between the intentions of both amateur and professional musicians and the audio manipulation tools available through software. Rather than force nonintuitive interactions, or remove control altogether, we reframe the controls to work within the interaction paradigms identified by research done on how audio engineers and musicians communicate auditory concepts to each other (Jones 1992; Porcello 2004). Specifically,

- Evaluation (e.g., "I like the first equalization setting better than the second one.")
- Descriptive words for sound (e.g., "Make it sound 'hollow"')
- Vocal imitation (e.g., vocally making a "whoosh" sound to illustrate a transition idea)
- Exploration (e.g., "What other things are like this, but different?")

In this article, we provide an overview of our research on building tools to support these kinds of interactions. We describe the learning algorithms, design frameworks, and software that support user communication of audio concepts through these interactions types. Although each of the tools described in this article has been described in its own prior publication, this overview article is the first to connect all these tools together in a single narrative that describes the underlying interaction paradigm, design philosophy, and guiding principles underlying all these works. This allows us to contrast the approach we advocate with that embodied in existing audio production tools and provide general guidelines to help future tool builders decide which approach is best for the tool they are creating.

These contributions provide a research foundation for a new generation of audio production tools and define approaches to building tools that help musicians realize their ideas with minimal effort and in their own language. We hope the work will inspire developers to make audio production tools that use natural interaction paradigms that empower, support and enhance human creativity.

2. SocialEQ: An Evaluative Interface

One of the first steps we took towards the goal of rethinking control of audio production tools was to replace the standard controls used for one of the most common audio effects tools: the equalizer. Equalizers affect the timbre and audibility of a sound by boosting or cutting the amplitude in restricted regions of the frequency spectrum. They are widely used for mixing and mastering audio recordings. Many have complex interfaces that lack clear affordances and are daunting to inexperienced users. For example, to make an audio recording "bright", a user cannot simply turn the "bright" knob on the parametric equalizer in Figure 1 to achieve their desired effect, as there is no such knob. Instead one must know the mapping between the concept (e.g., "bright", "tinny", and "warm") and the low level controls. This can be daunting when a tool has over 40 controls on it, like the equalizer in the figure.

Rather than force a user to learn the ins and outs of parametric equalization, we took inspiration from the way that vision prescriptions are determined. At no point does the end user (the future wearer of prescription eyewear) specify parameters of the lenses directly. Instead, a series of options is presented and each option is evaluated by the user. As the user gives evaluative feedback, the options are honed until a prescription that best facilitates vision is provided. This approach can be operationalized to work automatically by applying a technique similar to correlation-based techniques used in psychophysics (Calandruccio and Doherty 2007; Lutfi 1995). Such techniques estimate the relative perceptual importance of stimulus features by computing how strongly modifications to that feature correlate with some user-generated variable. We have done this and the overview of our approach is as follows:

1. The user selects an audio file, and a word to describe their auditory goal (e.g., "bright" or "warm").

2. We process the audio file once with each of N equalization settings. This makes N examples of the same audio, each with different equalization applied.
3. The user rates how well each example achieves the goal.
4. The system learns an equalization setting by estimating the effect of each frequency band on user response. This is done by correlating user ratings with the variation in gain of each band over the set of examples.
5. The system presents the learned equalization setting to the user, along with a slider that controls the extent to which that setting is applied.

We developed this approach to control an equalizer (Sabin and Pardo 2009, 2013) and commercial equalizer software based on this approach was released under the name *iQ*, by the company Ear Machine, founded by our collaborator, Dr. Andrew Sabin. The iQ equalizer was positively reviewed by Jon Burton in Sound on Sound magazine (Burton 2011), where he stated:

> It's a great idea for inexperienced engineers getting to grips with frequencies and equalization, as well as for nontechnical musicians...There will be a time in a session where you do not have the energy to find out what the guitarist actually means when they ask you to make it "more toppy but without adding treble". Drop iQ on to the channel, sit down with a cup of tea and a biscuit, and leave them to their own devices...

Figure 3 shows SocialEQ (Cartwright and Pardo 2013), a web-based tool for equalization. The interface for SocialEQ is essentially identical to iQ and both tools use the evaluative paradigm we described. The equalization setting learned in this figure was a user's definition for "bright". This setting was found without requiring the user to know how controls like "Q" would map onto this concept. Note that nothing in the evaluative interaction paradigm embodied by SocialEQ is specific to equalization and the same interaction paradigm can be applied to other controls.

Figure 3. SocialEQ: an evaluative interface for controlling an equalizer. (**left**) The evaluation panel. Here, the user rates options to teach the machine a concept. (**right**) Once the machine has learned a concept ("bright", in this case), a single-slider control is provided to let the user fine-tune the equalization in terms of that concept. Screen capture created by the authors.

To apply the underlying algorithm described earlier, the control parameters should ideally have a linear relationship with perception, and parameters should not have strong interactions between each other. This is due to the underlying paradigm used to learn an effective setting from user ratings of examples (Sabin et al. 2011). It learns linear relationships between user ratings and parameter settings. To learn nonlinear relationships, one would have to develop an alternative method, although the underlying interaction paradigm of evaluation of examples would remain the same. There has been some follow-on work that explores how to learn nonlinear relationships from user feedback on setting synthesizer parameters (Huang et al. 2014), so we believe this approach is generalizable. In this

article, however, we advocate for the interaction paradigm rather than arguing in favor of a particular algorithm used to learn from this interaction paradigm.

Used appropriately with its limitation in mind, the interaction approach these algorithms support (evaluation of options, rather than direct control) can, however, be a viable one in any situation where evaluating examples may be a more appropriate interaction than directly manipulating the underlying low-level controls and an algorithm can be applied to map user preferences to control parameters (Amershi et al. 2011). Once we established the effectiveness of this approach for equalization, we later applied the evaluative approach to control reverberation (Sabin et al. 2011), as this auditory effect conformed sufficiently to the aforementioned constraints that the algorithm from SocialEQ could be applied without modification.

Along with equalization, reverberation is one of the most widely used audio processing tools. Natural reverberation is caused by the reflections of a sound off of hard surfaces (e.g., walls), causing echoes to build up and then decay as the sound is absorbed by the walls and air. Artificial reverberation simulates this process and adding reverberation (reverb) to a sound gives the listener the impression that a sound occurred in a location (e.g., a canyon or a church) different from where it was recorded (e.g., a sound booth).

We performed a user study that showed the approach of controlling a reverberator by having a user rate example reverb settings applied to the sound required the user to rate 10 to 15 examples (Sabin et al. 2011). A similar study for equalization showed that equalization took longer, requiring between 20 and 25 rated examples (Sabin et al. 2011).

2.1. Using Prior Knowledge

The larger number of evaluations needed to find an effective equalization setting caused us to investigate approaches to reduce the required number of rated examples for equalization. We found that we could greatly reduce the number of examples that required evaluation if we could use prior knowledge learned from interacting with previous users (Kim and Pardo 2014; Pardo et al. 2012). To gain a base of prior interactions, we turned to the microtask labor market of Amazon Mechanical Turk. Amazon Mechanical Turk (MTurk) is a website where one can hire remotely located workers to perform discrete on-demand tasks. In our case, the task was equalizing a short musical passage using SocialEQ.

Via MTurk, we collected 3369 sessions of user data from SocialEQ. For each session, a user was presented an audio file and asked to freely select a word that would describe an audio effect they would like to achieve (e.g., make the sound "darker" or "fatter" or "warmer"). We then performed the interaction described earlier using the SocialEQ interface (see Figure 3). All participants rated equalization curves drawn from a shared set of examples used as a baseline set of standard equalization curves developed for this work (Sabin et al. 2011).

To use only high-quality prior user data, we inserted some repeat examples into each person's session. We then filtered out sessions from people who gave very different ratings to repeat examples. In other words, only sessions from people who put in effort and rated examples consistently were selected. This data filtering left us 1635 sessions.

We utilized the user data collected in these sessions to reduce the number of examples a user has to rate. The idea is that if two users are close to each other in terms of the ratings they gave to audio examples, the resulting EQ curves learned after all 25 ratings should also be similar, which we showed to be true in a validation study (Kim and Pardo 2014). When the current user rates the first n examples, we can measure similarities between ratings from the current user and all the prior users, and find the k most similar prior users to the current user.

Then, rather than have the current user rate all 25 example equalization curves, we can estimate the current user's ratings of currently unrated examples by taking a weighted average of ratings that the k prior users gave to those examples. In our work, we found that using the 64 prior users most similar to the current user worked best (Kim and Pardo 2014). The weights for the weighted mean of

prior user ratings are based on how similar each prior user is to the current user in terms of the ratings to the n examples.

To speed up the learning even more, we set the order of presentations of examples to the current user to most quickly determine which prior users provided the most similar answers to the current one. We do this by selecting the examples for the current user to rate that caused the greatest disagreement among prior users. This is a kind of query-by-committee active learning (Cohn et al. 1994). We selected this approach for simplicity and appropriateness to the task. A comparison to other approaches is outside the scope of this article.

Obtaining ratings to those informative examples greatly narrows the search space, speeding up the equalization learning. As a result, the updated SocialEQ only needs eight user-rated examples to achieve an acceptable equalization setting (Kim and Pardo 2014; Pardo et al. 2012). This is a reduction from the roughly 25 ratings required when access to prior user interactions is not available. Note that, to apply equalization, no end user knowledge of signal processing, frequency bands, Q, decibels or other technical terminology is needed. This provides a working example of a fundamentally new way of controlling an audio effect with evaluative feedback alone. This fundamentally new paradigm can offer a significant time savings for those whose primary goal is not to become experts in parametric equalization but rather to use the power of an equalizer as a step towards completing a task that requires some equalization but not the level of control or time to learn typically required by a tool like the one illustrated in Figure 1.

3. Using Descriptive Language

Given our ability to lower the number of rated examples required to create a useful equalization setting by using prior knowledge, we wondered if the interaction could be sped even further. A trained engineer will have a clear idea of what to do if a person says the music is too "bassy" (lower the gain on the low frequencies). This is because they have learned this mapping between a descriptive word ("bassy") and the action to be performed. If we could build a vocabulary of nontechnical words, paired with the actionable changes applied to the interface to embody those words, we could skip learning entirely and just let the user specify their desires in natural language.

Unfortunately, the number of descriptive words applied to sound is large and their mappings to actionable changes on effects tools had not been mapped. In fact, potential users of audio production tools (e.g., acoustic musicians, podcast creators) often have sonic ideas that they cannot express in technical terms with known mappings onto the low-level parameters of existing tools. Therefore, interactions between audio production professionals and these content creators can be a frustrating experience, hampering the creative process. As John Burton of Sound on Sound magazine put it (Burton 2011):

> ...how can you best describe a sound when you have no technical vocabulary to do so? 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, can only abstract. I have been asked to make things more 'pinky blue', 'Castrol GTX-y' and 'buttery'.

In our work on evaluative interfaces, we had learned how to crowdsource data from a large number of people. Therefore, we decided to directly learn a vocabulary of descriptive adjectives for audio from a group of nonexperts in audio engineering. Importantly, in this work we also learned the mapping from the descriptive term to actionable changes in the audio so that one could use this vocabulary to control audio effects produced by three of the most widely used effects tools: equalization (EQ), reverberation, and dynamic range compression (compression). Equalization adjusts the gain of individual frequencies in a recording and can be used to make things sound brighter or warmer. Reverberation adjusts the spatial quality of an audio recording by adding echo effects to the audio and can be used to make things sound like they were recorded in a cave, or a church, or a

stairwell, etc. Compression reduces the amplitude variability over time of an audio recording, so that soft sounds are boosted and loud sounds are reduced in gain.

The goal of this vocabulary collection was to learn a vocabulary that makes audio production interfaces accessible to laypeople, rather than experts. We did this by collecting vocabulary words from nonexperts in audio engineering and finding mappings between commonly used descriptive words (e.g., "boomy") and the hard-to-understand controls of production tools (e.g., "predelay"). The eventual goal of this work was to develop interfaces that give novices an easy point of entry into audio production, thus supporting the creativity of acoustic musicians without forcing them to learn interfaces with opaque and esoteric controls.

3.1. SocialFX: Collecting a Vocabulary

The vocabulary collected for equalization was done as part of the SocialEQ project, discussed in Section 2.1 of this article. When equalizing an audio file using SocialEQ, the user was asked to label the resulting EQ setting with a descriptive word of their choice that best embodied the effect of this equalization. SocialEQ learned 324 distinct words in 731 learning sessions (Cartwright and Pardo 2013).

The vocabulary for reverberation (Seetharaman and Pardo 2014) and the vocabulary for compression (Zheng et al. 2016) were both collected in a different way, as part of a project we called SocialFX. We sought to make the data collection more lightweight so that we could gather more words from more people. Therefore, instead of asking people to complete a task using the tool in question, we structured the data collection as follows.

For SocialFX, participants were recruited through Amazon's Mechanical Turk. Each participant was presented with a 10–15 s audio recording of a musical instrument (drums, guitar, or piano) recorded without any audio effect applied. Then, one effect (either reverb or compression), with a randomly selected setting of parameters, was applied to the audio. The participant was provided a 'Effect On/Off' button, letting them turn an audio effect on and off to hear how it modified the sound.

Once a participant had toggled the effect on and off and listened for at least another full iteration of the audio example, the participant was asked to provide a list of words that describe the effect. We encouraged single words, like "warm" and "spacious", but also allow multiword descriptions. Once the participant had finished describing the effect, a set of 15 other words was shown to them. These words were drawn from the vocabulary of audio effects words provided by previous participants. The participant was then asked to check the box next to any word that they felt described the effect they had just heard. This let us confirm how widely a particular word was applied to an effect setting. It also let us see how general or specific the term was. For example, the word 'echo' was selected as a word for practically every reverberation setting, while 'underwater' only applied to a specific subset. The interface for the first stage of the data collection is shown in Figure 4.

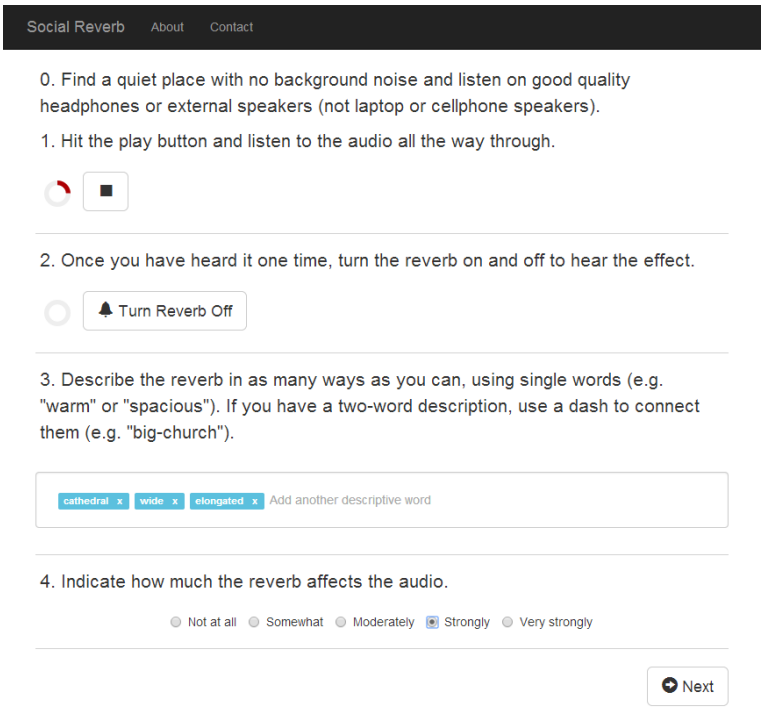

Figure 4. SocialReverb: Participants are asked to listen to a dry recording, then a recording with an audio effect applied, and then describe it in their own words. In the second stage, they are asked to check off words that other participants contributed that they agree also describe the audio effect. Screen capture created by the authors.

In collecting a vocabulary for reverberation, we performed a total of 1074 sessions, in which we collected 14,628 word instances from 513 people. These 14,628 word instances represented 2861 unique descriptors of sound. Of the 2861 descriptors, 1791 had at least two instances. For compression, our data was collected from 239 individuals describing 256 unique instances of compression parameter configurations, resulting in 1112 unique descriptors. The data is available online[3] . A set of specific words that tend towards different effects or to general audio descriptors can be seen in Table 1.

Table 1. Descriptors and which audio effect they are related to. General words are used to describe audio effects produced by any of the three effects tools. Tending words are ones which were shown predominantly for a single audio effect, but appear in other audio effect vocabularies with low frequency. Specific words are ones that are used for a single audio effect and no others. The words shown above were found via inspection of the shared descriptor space between the three audio effects.

Word Category	Equalization	Reverberation	Compression
General words	warm, loud, soft, happy, cool, clear, muffled, sharp, bright, calm, tinny		
Tending words	cold, happy, soothing, harsh, heavy, beautiful, mellow	distant, deep, hollow, large, good, grand, spacey	quiet, full, sharp, crisp, energetic, subtle, clean, fuzzy
Specific words	chunky, wistful, punchy, mischievous, aggravating	haunting, organ, big hall, church-like, concert, cavernous, cathedral, gloomy	volume, sharpened, feel-good, rising, peppy, easy-going, earthy, clarified, snappy

[3] http://music.eecs.northwestern.edu/data/socialfx/

3.2. Audealize: A Word Cloud Interface to Control Effects

Most audio effects tools have drop-down lists of "presets" (saved settings) that are indexed by natural language terms. Given this, what need was there for us to create a new natural language-based interface?

Consider reverberation. We performed an analysis of the vocabulary we collected for reverberation and for equalization to the words in preset labels from the tools available in two commercial audio workstations: Adobe Audition and Ableton Live. For reverberation, only 15 words are shared between all three vocabularies: Ableton (69 words), Audition (120 words) and our SocialFX vocabulary for reverberation (369 words). Between Ableton and Adobe, only 24 words are shared. If we combine the Ableton and Adobe vocabularies and call it the "preset" vocabulary, just 38 words are shared between presets and the nonexpert population. This is just 10.2% of the total vocabulary used by the non experts. This percentage is representative for preset names applied to equalization and compression. Figure 5 illustrates the overlap in vocabularies between experts and nonexperts.

Existing commercial presets use words that are mostly distinct from the vocabularies of non-engineers. Further, the organization of the drop-down lists is typically alphabetical and the user has no understanding of the relationship between any two settings. This makes preset lists nearly as difficult to use as the interfaces they supplement. Note also that the preset vocabularies selected by the (presumably expert) developers of both the Ableton and Adobe reverberators showed little overlap (see Figure 5). This is indicative that many words selected by individual experts to describe reverberation are particular to the individual in question. This is another argument for bypassing expert vocabulary in tool building and, where possible, collecting vocabulary at a large-scale so that a set of generally agreed upon terms can be established. If this vocabulary were to be one that the general public understands and could use directly, so much the better. This would eliminate the need to teach new users definitions of acoustic terms that are specific to a small group of experts.

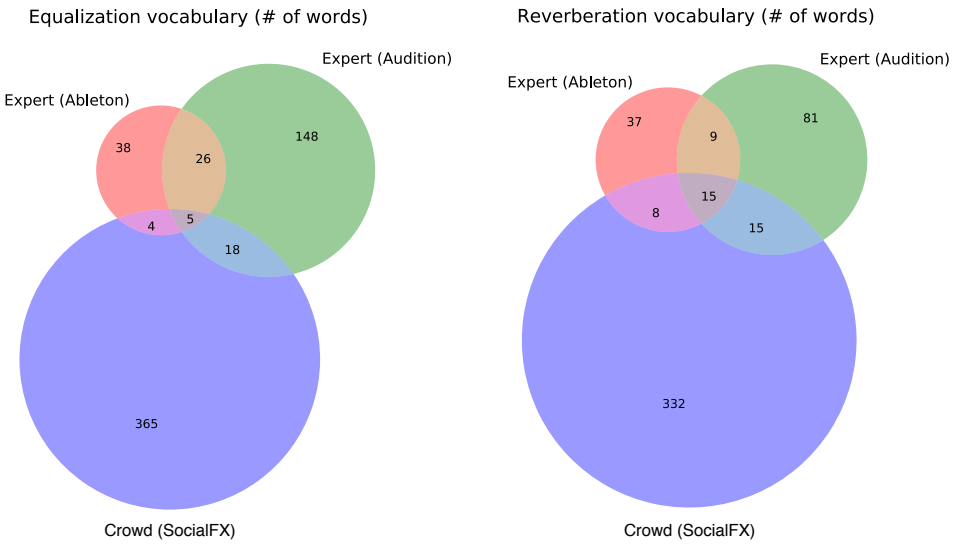

Figure 5. Overlap between three sets of vocabulary for equalization (**left**) and reverberation (**right**)—preset vocabulary from Ableton (73 EQ words, 69 reverb words), Audition (197 EQ words, 120 reverb words), and the vocabulary used by a nonexpert population (394 EQ words, 369 reverb words). Figure created by the authors.

Given that perspective, a better interface would use words nonexperts choose, that are validated by others and that places the words in some organization where the relationships between them are evident.

Using the vocabulary we collected in the SocialFX project, we created Audealize, an interface for reverberation and equalization. Audealize is based on natural language adjectives a nonexpert

would use to describe the audio effects. The user modifies a sound recording by clicking on descriptive terms (e.g., "warm" and "tinny") in a two dimensional word-map interface to control production tools, where the distances between words correspond to the similarity of the effects the words describe. 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. We created a two dimensional word-map reverberation and one for equalization. The Audealize interface (and word map) for equalization is shown in Figure 6. Note that our SocialEQ data collection solicited data from both English and Spanish speakers (Cartwright and Pardo 2014b), so both languages are displayed.

Recall that each word in the vocabulary for a production tool has a mapping to the parameters required to elicit that word. We built each map using multi-dimensional scaling (Kruskal and Wish 1978). Multi-dimensional scaling is a method to project a higher dimensional space into a lower dimensional one, by preserving the distances between items in the space. For reverberation, we mapped a five dimensional feature space (the five parameters controlling the reverberator) to a two dimensional space. For equalization, we map a 40-dimensional space (40 frequency bands) down to a two dimensional space. The result is a 2-dimensional map for each effect where closely related effects are placed near one another. One advantage of using a word-map is that if a word is unfamiliar to the user, a familiar word nearby the unfamiliar word can serve as an auditory synonym and guide the user. For example, in the word-map presented in Figure 6, "flat" may be unfamiliar but it is near "dulling", which indicates the effect it may have on the sound. Alternatively, "vibration" in the upper left may be ambiguous in the context of reverberation word but the fact that it is near the words "high-pitch, stadium, dissonant" provides some indication of the type of effect it will have on the sound.

It is worth noting that the approach of selecting a descriptive word to set the reverberation or equalization can help save time for more experienced users of these tools. For example, one can get close to the desired settings quickly by selecting a high-level descriptive word that approximates the goal (e.g., use the Audealize interface to select a "bright" sound), at which point an experienced user can precisely control the tools using original parameter space controls. This could even be a two-person process, the inexperienced musician could select a word to approximate their goal. The experienced engineer could then take over, fine tuning as appropriate.

Note that this word-based approach is appropriate and works well for the kind of manipulations it was intended for: mapping general describe-able goals to actionable changes in the sound. It is not an appropriate interface for precise settings whose details are based on a particularity of the input sound. For example, if one were to need a notch filter to remove an annoying hum at 185 Hz in a particular recording, an interface based on natural language (remove that annoying hum) would not be effective and the precise control of a typical parametric equalizer would be called for. Using Audealize, we performed the first user study comparing a word-map interface, like the one in Figure 6 to traditional audio production interfaces for reverberation and equalization (Seetharaman and Pardo 2016)[4]. We asked a population of 432 nonexperts to perform a match effect task using either the word-map interface or a traditional interface for the production tool in question.

In the match effect task, we asked study participants to listen to an audio clip that had either reverberation or equalization applied to it. We then asked them to use to manipulate a dry version of the same audio clip (one with no effect applied) to match the audio effect setting as closely as possible. They were given the correct effect to perform the match, with the only variation being whether the interface was a traditional one or the Audealize word map. We then measured how close each participant got to the actual effect in terms of Euclidean distance between features of the target reverberation or equalization setting and the user-provided setting.

---

[4] As of this writing, we have not yet implemented compression as an Audealize map.

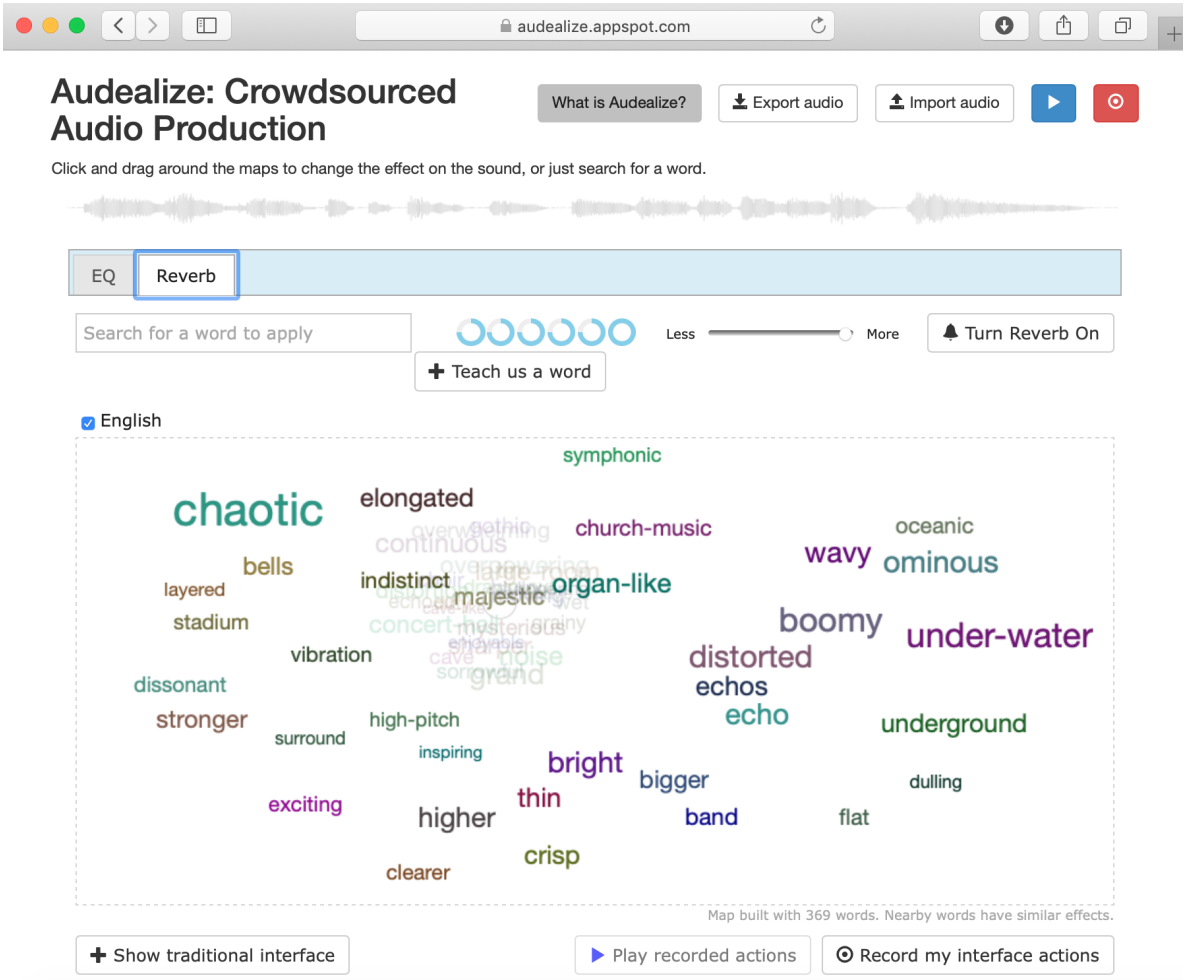

Figure 6. Audealize is a new way of looking at audio production tools. Instead of the traditional complex interfaces consisting of knobs with hard-to-understand labels, Audealize provides a semantic interface. Simply describe the type of sound you're looking for in the search boxes, or click and drag around the maps to find new effects. Screen capture created by the authors.

The results of the match effect task are shown in Figure 7. We observe that word-map interfaces are effective for match effect for equalization in all cases and for reverberation in easy cases. We also found that participants generally preferred the word-map interface to the traditional interface. We refer the reader to the paper (Seetharaman and Pardo 2016) for more details.

These results indicate a meaningful word-map visualization is an effective interaction paradigm for audio production by nonexperts. Audealize is an example of a general design approach that can apply to other domains. Additionally, the evaluative techniques discussed in Section 2 are incorporated into Audealize, accessible through the "Teach us a word" button. A traditional interface is also available for fine-tuning of the audio effect. Audealize can be tried in recent versions of Google Chrome at https://audealize.appspot.com/.

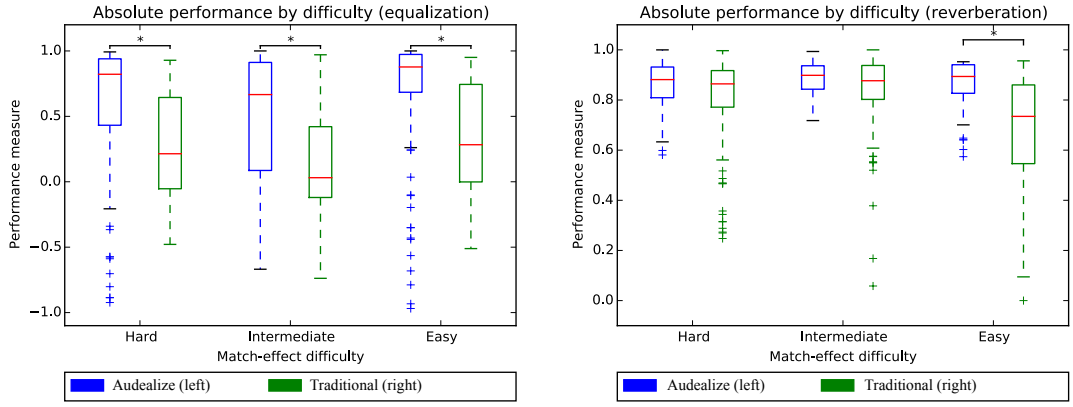

Figure 7. Match-effect test: absolute performance for equalization (**right**) and reverberation (**left**) broken down by difficulty. The * indicates statistical significance between distributions at a p-value < 0.01. We find significant difference in favor of the word-map interface used in Audealize in the easy task for reverberation and in all tasks for equalization. $N = 108$ for each box plot. Higher is better in each plot. Figure created by the authors.

4. Mixploration: Combining Evaluation and Exploration

One thing that any tool for artistic creation should do is to support exploration of potential alternatives. In general, things that are easy to discover are more likely to be applied. Every interface makes some kinds of exploration easy and others more difficult, inherently channeling creativity in certain directions. Therefore, it is reasonable to examine existing tools to see what kinds of exploration they support. In our work, we began looking at this issue through the lens of audio mixing.

Mixing refers to processing and combining multiple audio recordings (tracks) together into a single recording (the mix). Mixing is an essential step for many kinds of audio visual projects. A typical film, for example, mixes independent tracks for dialog, sound effects, environmental sounds and background music. Mixing consists of applying gain (a change in volume) to each track and summing all tracks together into the mix. In addition, there may be additional effects (e.g., equalization, reverberation, compression) added to each track. The standard mixing interface for this is shown in Figure 2, where there is one controller (e.g., fader) per track which controls the gain applied to that track, along with knobs to alter effects like equalization. Assuming a fixed number of audio effects, a mix at single point in time can be described as a point in an N-dimensional space, in which each of the N dimension represents a different mixing parameter. For example, a recording of a jazz trio might have three tracks (saxophone, piano, and bass) and a shared reverberation effect applied to all three tracks, but in different amounts per track (referred to as the per-track wet/dry mix). The gain of a track and wet/dry mix on each track gives a total of six dimensions.

Consider how a musical artist might explore this six-dimensional space using a conventional mixer interface. Typically, they will set the faders to an initial position (probably one of roughly equal gain for all tracks) and then move one fader at a time to improve the mix. This is a form of exploration where only a single dimension is varied at a time. This makes it difficult to explore the effect of changes brought about by changing multiple dimensions in parallel. In the case of mixing, this situation can result in a sub-optimal mix that sounds good but not great, and overall this approach may miss artistically satisfying alternatives that the artist simply did not explore, due to the nature of the controls.

Mixploration (see Figure 8) is a mixing interface to facilitate the discovery of diverse, high-quality rough mixes (i.e., they may need fine-tuning) through high-level exploration of the mixing space. Instead of the standard interface with one dedicated slider/knob controlling the volume or an equalization parameter of each track, the interface consists of a two-dimensional map, with each point on the map representing some setting of the gain and equalization parameters of all the audio tracks. With Mixploration, the user modifies the mix by moving around the map, changing

multiple parameters at once. Created using self-organizing mapping (Kohonen 1990), the map is a two-dimensional reduction of the high-dimensional level and equalization parameter space—it broadly covers the space of possible mixes, letting the user quickly move to very different points in the mixing space using a single control. At any point in the map, the user hears the resulting audio, without seeing the individual parameter settings. This is done to encourage the user to trust their ears and listen to the whole mix, rather than trusting their eyes and focusing on individual parameter settings. To encourage explicit evaluation of alternatives, the interface incorporates evaluation of mixes directly into the interface, allowing users to rate each point on the map with their keyboard while navigating the map using their mouse. When using Mixploration, people are encouraged to re-rate mixes as their preferences become more defined. The rating process can help the user to remember preferred mixes and concretize preferences, as well as aid the user in transitioning from divergent thinking (exploring the diversity of mixes in the two-dimensional map) to convergent thinking (concretizing a specific mix idea).

In Mixploration, we mapped an eight-dimensional parameters space to two dimensions. However, as the dimensionality of the parameter space increases, the map will become more erratic and possibly seem random. Therefore, if larger parameter spaces need to be explored, we recommend breaking them into groups of no more than eight parameters.

While this coarse map of the space encourages high-level exploration, it does not allow for fine-tuning the mix. Therefore, once a user rates several mixes and picks one *favorite mix*, the machine uses the ratings the user provided, along with their corresponding mix parameters, to learn a weighting function of what the user finds important. This approach is similar to the equalization learning approach taken in Section 2. For each individual control parameter, we perform a separate least-squares linear regression between the mix parameter values and the user's ratings of the mixes. We group the learned coefficients of the gain and equalization parameters into their respective weight vectors, and provide the user with a two-dimensional controller for them to refine the mix (see Figure 8, right).

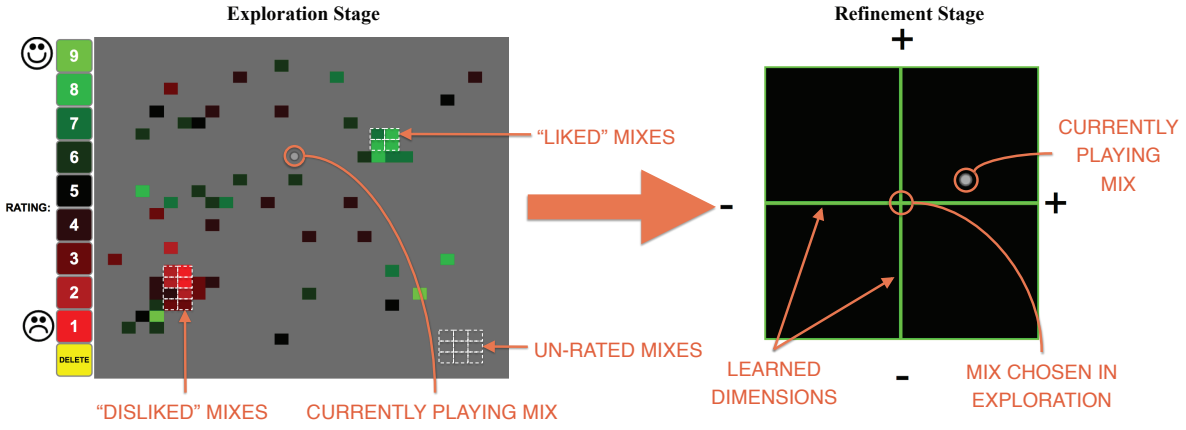

Figure 8. Mixploration is a tool for finding diverse mixes that separates the mixing process into two stages: exploration and refinement. (**left**) In the exploration stage, the user navigates the space of mixes on a 2D grid and rates how much they like the mixes they hear. (**right**) In the refinement stage, the user refines their favorite mix using a 2-dimensional slider, in which the dimensions are learned from the ratings in the exploration stage. Figure created by the authors.

To determine whether Mixploration achieved its goals, we conducted a 12-participant user study, comparing Mixploration to a traditional mixing interface with a gain (slider control) and two-band equalization (two knobs) for each track. Participants were asked to create three different mixes with each interface—a different song for each interface, with each song consisting of 4 instrument tracks. From the study, we found participants' mix satisfaction was similar for both interfaces. While participants preferred the traditional interface for precise mixing, participants preferred Mixploration for explorative mixing, and they created more diverse mixes and explored a larger

portion of the mixing parameter space with Mixploration (Cartwright et al. 2014). This indicates that an approach that allows exploration by moving in directions through the design space that are not easily supported by the traditional interface can be an effective way of allowing exploration of viable alternatives that might be difficult to discover using the standard interface.

5. SynthAssist: Vocal Imitation, Evaluation and Exploration

While our research has shown that descriptive language can work well for communicating some audio concepts, studies have shown that *vocal imitations* are more effective for communicating audio concepts that are unidentifiable or synthesized abstract sounds (Lemaitre and Rocchesso 2014; Lemaitre et al. 2013a, 2013b)—similar to how a sketched drawing may communicate an abstract visual concept more easily than a sentence can. Similarly, vocal imitation can also be thought of as sketching in the auditory domain (i.e., "auditory sketching")—a quick, rough approximation of an audio idea that is accessible to many people.

The use of vocal imitation for communicating and/or sketching audio concepts between humans has been observed in a variety of audio contexts. For example, an anthropologist observed experience and novice recording engineers communicating various snare sounds to each other by imitating them with their voice, e.g, "<*tsing, tsing*>", "<*bop, bop, bop*>", "<*bahpmmmm, bahpmmmm, bahpmmmm*>", "<*kunk, kunk*>", "<*dung, gu kung (k) du duku kung*>", "<zzzzz>", and "<*pts*>" (Porcello 2004). Even formally educated composers like Stanford professor Mark Appelbaum use vocal imitation to communicate composition ideas—as can be heard in his piece Pre-Composition (2002) (Appelbaum 2003), a composition that provides a glimpse into the composer's creative process. Lastly, through a workshop study, Ekman et al showed that vocal imitations can be an effective method for sketching and designing sonic interactions (Ekman and Rinott 2010). SynthAssist (see Figure 9) is a system that enables users to program a synthesizer using vocal imitation and evaluative feedback. It uses a data-driven audio-retrieval approach in which synthesized audio samples and their associated audio features and synthesis parameters are queried using vocal imitation and both the query and the distance function are refined and adapted using evaluative feedback. SynthAssist has a database of thousands of entries, each of which contains a synthesizer setting (i.e., a parameter-based model), an example audio recording generated with the synthesizer setting, and the synthesizer setting model (i.e., an audio feature-based model) on which each entry is keyed. Through interactions with the user, the system builds a model of the user's desired sound and uses this model to query to the database and retrieve synthesizer settings that are similar to the desired sound model. Therefore, the success of this approach relies on the system's ability to model the desired sound (given an input example, e.g., vocal imitation and evaluative feedback from the user), synthesizer sounds, and a similarity measure between these models.

To model both the synthesizer sounds and the desired sounds, we extract time series of a small number of high-level features from the audio. We then represent the sounds using these time series as well as the time series standardized to themselves. The motivation of this approach is that while our voice has a limited pitch range and timbral range, it is very expressive with respect to how it changes in time. For example, your voice may not be able to exactly imitate your favorite Moog bass sound, but you may be able to imitate how the sound changes over the course of a note (e.g., pitch, loudness, brightness, noisiness, etc.). Therefore, these models focus on how the sound changes through time. We measure how similar two models are by calculating the dynamic time warping (DTW) similarity (Sakurai et al. 2005) between their time series. We compute DTW on each feature independently, and then we linearly combine the resulting similarity scores weighted by the relevance of each feature learned through the user feedback. Using this feedback, we also refine the desired sound model by creating an alignment-informed weighted average of synthesized setting models the user rated as relevant (Niennattrakul and Ratanamahatana 2009).

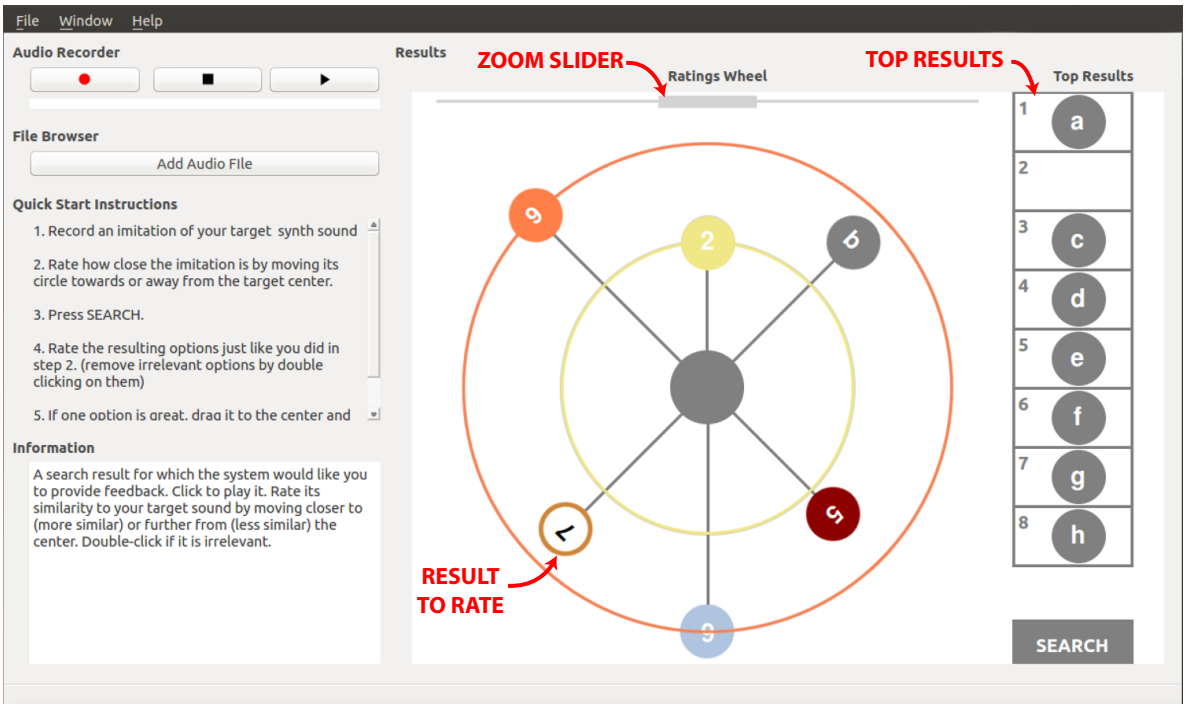

Figure 9. SynthAssist: A new way of programming a music synthesizer. The user starts by recording or uploading a sound similar to what is desired. The interface explores the settings of the synthesizer and returns settings most similar to the example along different dimensions. Each suggestion is represented by one of the colored circles. When a user clicks or moves a suggestion, the synthesizer sound changes. Users rate how similar suggestions are to their target by moving them closer to or farther from the center. If a suggestion is irrelevant, the user can inform SynthAssist and remove it from the screen by double clicking on it. Based on this feedback from the user, the system refines its model of the desired sound and suggests more settings. Dragging a suggestion to the center of the circle indicates this is the desired sound and terminates the interaction. Figure created by the authors.

To evaluate this system's interaction paradigm and affordances for novices, we ran a user study with 16 participants that were pre-screened to have minimal experience with synthesizers (Cartwright and Pardo 2014a). In the study, we compared SynthAssist to a traditional synthesizer interface consisting of knobs and buttons. Figure 10 show an example of a synthesizer that embraces this interface paradigm. We performed this comparison by asking the participants to use the interfaces to match the synthesizer's output to target sounds. On average, SynthAssist enabled "novice users" to program the synthesizer to produce sounds closer to the target sounds than the traditional interface—and more quickly. Participants overall felt that they were able to achieve their desired sound more easily with SynthAssist and would recommend the software to novices. However, there is still room for improvement with the software since there were times that the software did not do what they wanted it to do, and we also discovered that some audio concepts were much more easily communicated to the system than others using vocal imitation.

Figure 10. The Novation X-station 49 synthesizer was produced between 2004 and 2009. The design embraces the knobs, buttons and sliders control paradigm we seek to provide an alternative to. This image is by deepsonic[5] and is licensed under the CC BY-SA 2.0[6] license.

While a primary goal of SynthAssist was to support vocal imitation input, the methods used to model and compare sounds are not limited to vocal imitations. Therefore, a user could also provide SynthAssist with an initial recording of a sound that has some but possibly not all of the characteristics of a desired synthesizer sound. The interaction paradigm supported by SynthAssist is a mixed-initiative one, where the user provides examples, the system returns options and they work together to reorganize the space of possibilities so that the exploration of the space is increasingly along the dimensions that matter to the user, as the interaction continues. This is a profoundly different interaction than the one embraced by the designers of traditional synthesizers like the Novation X-station (see Figure 10). We expect the approach embodied by SynthAssist will appeal to a different set of users than those who prefer the traditional approach. By building tools that embrace this new interaction paradigm, we expect that a broader group of people will be encouraged and enabled to contribute creatively in this domain.

6. Conclusions

In this article, we have provided an overview of our work in applying a user-centered design approach to the development of adaptive audio production tools. This work can be characterized as part of the paradigm shift in human-centered computing toward the design and development of natural user interfaces. The tools automatically adapt to the user's conceptual framework by learning from the user, rather than forcing the user to adapt to the tools. Where appropriate, the tools speed and enhance their adaptation using active learning informed by interaction with previous users (transfer learning). These interactions include providing examples, evaluative feedback and exploring the space of possibilities in new ways. We also showed how interfaces can be built from crowdsourced data into an audio concept map, grounded in user percepts. These concept maps allow people to control production tools using nontechnical language that more directly aligns with how many people in the broader community describe audio.

---

5 https://www.flickr.com/photos/73143485@N02/39606545252
6 https://creativecommons.org/licenses/by-sa/2.0/

Some may ask why we have worked to create control paradigms (evaluative feedback, providing examples, natural language descriptors) that do not require specific knowledge of tools or technologies. Wouldn't it be simpler to just learn the existing tools and be done with it? Consider musical instruments as an analogy. A violin provides very detailed, moment-by-moment pitch and timbre control based on how one places one's fingers on the strings and the pressure and position of the bow. This comes at a cost, however. It takes most people many months of effort to produce a pleasing tone on the violin, and it takes much longer to produce the range of pitches and timbres the instrument is capable of. A well-tuned piano lets even a beginner produce a pleasing tone with a simple key press. While this means giving up the fine-grained pitch control possible with a violin, it allows for different kinds of exploration of the sonic space as the musician is free to focus on other aspects of creation (e.g., complex harmonies, multiple melodic lines).

It is true that many users have the time, energy, and interest needed to learn the specifics of a variety of audio processing tools and interfaces. But for many others, the ability to simply use a word they know (e.g., make it "warm") or provide evaluative feedback on a small number of examples sidesteps the issue of having to learn specific tools, terminology and technologies. Consider a low-budget video producer who can't afford an audio engineer. Such tools as we describe may let them solve an immediate problem (e.g., making the voice sound less "boomy") and move on to the next task in their project, where they may wish to express more fine-grained control (selection of the right camera angle from a set of alternate shots). Interfaces that provide a middle ground between investing the time to become an expert engineer and relinquishing all control can be very helpful to such artistic creators.

One take-away we hope to impart to the reader is that there are a number of control paradigm choices available to the tool builder and that there are many situations where the standard control paradigms inherited from hardware controllers may not be the best for all tools or all users. As with all control paradigms, the ones we have presented tend to make certain things easier and other things harder. The trade-off for any tool-builder is to decide who their intended users are and select an interface appropriate for the task and the user. If one expects the user to be someone who wants to achieve a general goal (e.g., making the sound more "bright"), does not want to dedicate hours to learning an interface or tweaking a result, then a descriptive language interface (e.g., word-cloud controller) may be very appropriate. If that person has a specific idea in mind, but cannot articulate it in words, then using examples and evaluative feedback may be a more appropriate choice of interface. If someone wants to really learn to do fine-grained control of an audio tool, grounded in an understanding of the underlying technology, then a traditional interface in which all of the low-level parameters are visible and controllable (e.g., the parametric equalizer in Figure 1) may be a more appropriate choice.

The example tools we have created help define approaches to building tools that help musicians realize their ideas with minimal effort and in their own language. We hope these tools will provide examples and the approaches will be more broadly adopted to provide facilities for computer-aided, directed learning, so that creative artists in audio production can expand their conceptual frameworks and abilities.

There is some evidence that the approaches we have presented are beginning to be adopted by commercial tool builders in the space of music creation. In particular, example-based control is being adopted. For example LANDR's fully automated mastering tool[7] now allows users to upload an example audio file that is mastered as the user would like their new track to be mastered. The tool's mastering procedure now attempts to match sonic characteristics of the provided example. Google's Project Magenta[8] has a plug-in that can be used with any DAW. This plug-in is used to generate

[7] https://www.landr.com.
[8] https://magenta.tensorflow.org.

musical passages (as opposed to performing mixing or adding effects). As with LANDR, example files can be provided to Magenta's Interpolate function to guide the range of musical passages that may be output by the system. It can then generate a set of interpolated examples that morph between pairs of user-provided examples.

Natural language-based interfaces crowdsourced from data have yet to be adopted widely by industry, but there are some examples of tools that attempt to use broadly known terms to control musical or audio processing tools. Apple's Logic Pro Digital Audio Workstation (DAW) now provides an automated drummer that is controlled on a two-dimensional plane where the horizontal axis is labeled simple/complex and the vertical axis is labeled soft/loud. The drumming style of the automated drummer is controlled by moving a dot around this plane to produce different drum patterns. This is an example of high-level language control using commonly-understood words.

To-date, iQ (described in Section 2) is the only mixing, mastering or audio effects tool we are aware of that explicitly uses evaluative feedback of examples to control the tool. This kind of interface has, however, been adopted in the area of hearing assistive devices. Bose, in particular has purchased the rights to the patent for equalization preference learning embodied in iQ and SocialEQ. This bodes well for the dissemination of evaluative feedback in the space of consumer devices for amplification and reproduction of audio.

While we have established these natural interaction paradigms as alternative interactions for working with audio production tools, the methods that we have presented to achieve these interactions do have some limitations. For example, Audealize, SocialEQ, and SocialFX are not adaptive to the input signal. This may reduce their ability to produce certain audio concepts when applied to specific input signals. In addition, the mappings for these tools were learned for individual audio processors (e.g., equalization, reverberation, compression), but some audio concepts may require a combination of audio processors in which a mapping is jointly learned (e.g., "boxy" may require a combination of equalization, compression, and reverberation to produce the desired effect). These approaches also do not adapt to the presence of other signals in a mix, yet we know from psychoacoustics that the presence of other signals affects perception (Moore et al. 1997). Lastly, Mixploration and SynthAssist have not been tested with higher-dimensional parameter spaces and will likely require modifications or alternative mapping strategies to work with large numbers of parameters. Future work in this area should address these limitations.

The primary purpose of our research into the interaction paradigms we describe in the article is to develop alternatives to the two main approaches that still dominate the audio production design space: low-level controls and full automation with little or no control provided to the user. That said, another natural extension of the work described in this article is to incorporate evaluative, exploratory and language-based interfaces into teaching tools that scaffold users to learn the traditional low-level parametric interfaces. Evaluating examples and seeing the parameter settings chosen by the interface in response to natural language descriptors naturally illustrate connections between parameter setttings, the language people use to describe sound and evaluative choices users make. Incorporating these interaction approaches into educational tools may prove another fruitful area for future research.

As a final note, we do not advocate that those who are happy with the existing tools stop using them. We do advocate, however, for the creation of a new class of tools based on the interactions we have presented and we expect that doing so will broaden participation in the arts to groups of people for whom the existing tools are not a good fit. By providing tools that let people manipulate audio on their own terms and enhance their knowledge of such tools with directed learning, we hope to transform the interaction experience for such people, making the computer a device that supports and enhances creativity, rather than an obstacle.

7. Patents

The work described in this manuscript resulted in the following patents.

Systems, methods, and apparatus for equalization preference learning US8565908B2.

Systems, methods, and apparatus to search audio synthesizers using vocal imitation US9390695B2.

Author Contributions: Author contributions are approximately as follows: conceptualization, B.P.; methodology, B.P., M.C., P.S., B.K.; software, M.C., P.S., B.K.; validation, B.P., M.C., P.S., B.K.; formal analysis, M.C., P.S., B.K.; investigation, B.P., M.C., P.S., B.K.; resources, B.P., M.C., P.S., B.K.; data curation, B.P., M.C., P.S., B.K.; writing—original draft preparation, B.P., M.C., P.S., B.K.; writing—review and editing, B.P., M.C., P.S., B.K.; visualization, M.C., P.S.; supervision, B.P.; project administration, B.P.; funding acquisition, B.P.

Funding: This research was funded, in part, by the National Science Foundation, grant numbers 1116384 and 0757544.

Acknowledgments: We thank Andy Sabin for his key contributions in developing the algorithm for evaluative equalization used in the iQ and SocialEQ equalizers. We thank Taylor Zhang for his contribution to data collection of descriptive terms for dynamic range compression. We thank Olivia Morales and Zach Puller for their work on the interface for Audealize. We thank Ethan Manilow and Abir Saha for providing feedback on this submission.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

Amershi, Saleema, James Fogarty, Ashish Kapoor, and Desney Tan. 2011. Effective end user interaction with machine learning. Paper presented at Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, August 7–11.

Appelbaum, Mark. 2003. Pre-composition. Available online: https://www.youtube.com/watch?v=Tlx4DK_h8Ws (accessed on 28 August 2019).

Burton, Jon. 2011. Ear Machine iq: Intelligent Equaliser Plug-in. Available online: http://www.soundonsound.com/sos/jun11/articles/em-iq.htm (accessed on 28 August 2019).

Calandruccio, Lauren, and Karen A Doherty. 2007. Spectral weighting strategies for sentences measured by a correlational method. *The Journal of the Acoustical Society of America* 121: 3827–36. [CrossRef] [PubMed]

Cartwright, Mark, and Bryan Pardo. 2014a. Synthassist: An audio synthesizer programmed with vocal imitation. Paper presented at 22nd ACM international conference on Multimedia, Orlando, FL, USA, November 3–7, pp. 741–42.

Cartwright, Mark, and Bryan Pardo. 2014b. Translating sound adjectives by collectively teaching abstract representations. Paper presented at Collective Intelligence Conference, Cambridge, MA, USA, June 10–12.

Cartwright, Mark, Bryan Pardo, and Josh Reiss. 2014. Mixploration: Rethinking the audio mixer interface. Paper presented at 19th international conference on Intelligent User Interfaces, Haifa, Israel, February 24–27, pp. 365–70.

Cartwright, Mark Brozier, and Bryan Pardo. 2013. Social-eq: Crowdsourcing an equalization descriptor map. Paper presented at 14th International Society for Music Information Retrieval, Curitiba, Brazil, November 4–8, pp. 395–400.

Cohn, David, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning* 15: 201–21. [CrossRef]

Ekman, Inger, and Michal Rinott. 2010. Using vocal sketching for designing sonic interactions. Paper presented at ACM Conference on Designing Interactive Systems, Aarhus, Denmark, August 16–20, pp. 123–31.

Huang, Cheng-Zhi Anna, David Duvenaud, Kenneth C Arnold, Brenton Partridge, Josiah W Oberholtzer, and Krzysztof Z Gajos. 2014. Active learning of intuitive control knobs for synthesizers using gaussian processes. Paper presented at 19th international conference on Intelligent User Interfaces, Haifa, Israel, February 24–27, pp. 115–24.

Jones, Steve. 1992. *Rock Formation: Music, Technology, and Mass Communication*. Foundations of Popular Culture. Newbury Park: Sage.

Kim, Bongjun, and Bryan Pardo. 2014. Speeding learning of personalized audio equalization. Paper presented at 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, December 3–6, pp. 495–499.

Kohonen, Teuvo. 1990. The self-organizing map. *Proceedings of the IEEE* 78: 1464–80. [CrossRef]

Kruskal, Joseph B., and Myron Wish. 1978. *Multidimensional Scaling*. Thousand Oaks: Sage, vol. 11.

Lemaitre, Guillaume, and Davide Rocchesso. 2014. On the effectiveness of vocal imitations and verbal descriptions of sounds. *The Journal of the Acoustical Society of America* 135: 862–73. [CrossRef] [PubMed]

Lemaitre, Guillaume, Patrick Susini, Davide Rocchesso, Christophe Lambourg, and Patrick Boussard. 2013a. Using vocal imitations for sound design. Paper presented at International Symposium on Computer Music Multidisciplinary Research, Porto, Portugal, September 25–28.

Lemaitre, Guillaume, Patrick Susini, Davide Rocchesso, Christophe Lambourg, and Patrick Boussard. 2013b. Non-verbal imitations as a sketching tool for sound design. Paper presented at International Symposium on Computer Music Modeling and Retrieval, Marseille, France, October 15–18, pp. 558–74.

Lutfi, Robert A. 1995. Correlation coefficients and correlation ratios as estimates of observer weights in multiple-observation tasks. *The Journal of the Acoustical Society of America* 97: 1333–34. [CrossRef]

Moore, Brian CJ, Brian R Glasberg, and Thomas Baer. 1997. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society* 45: 224–40.

Moorefield, Virgil. 2010. *The Producer as Composer: Shaping the Sounds of Popular Music*. Cambridge: Mit Press.

Niennattrakul, Vit, and Chotirat Ann Ratanamahatana. 2009. Shape averaging under time warping. Paper presented at International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Pattaya, Thailand, May 6–9 .

Pardo, Bryan, David Little, and Darren Gergle. 2012. Building a personalized audio equalizer interface with transfer learning and active learning. Paper presented at Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, Nara, Japan, November 2, pp. 13–18.

Porcello, Thomas. 2004. Speaking of sound language and the professionalization of sound-recording engineers. *Social Studies of Science* 34: 733–58. [CrossRef]

Sabin, Andrew T., and Bryan Pardo. 2009. A method for rapid personalization of audio equalization parameters. Paper presented at 17th ACM international conference on Multimedia, Vancouver, BC, Canada, October 19–24, pp. 769–72.

Sabin, Andrew Todd, and Bryan A Pardo. 2013. Systems, Methods, and Apparatus for Equalization Preference Learning. US Patent 8,565,908, October 22.

Sabin, Andrew Todd, Zafar Rafii, and Bryan Pardo. 2011. Weighted-function-based rapid mapping of descriptors to audio processing parameters. *Journal of the Audio Engineering Society* 59: 419–30.

Sakurai, Yasushi, Masatoshi Yoshikawa, and Christos Faloutsos. 2005. Ftw: Fast similarity search under the time warping distance. Paper Presented at Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Baltimore, MD, USA, June 13–17, pp. 326–37.

Seetharaman, Prem, and Bryan Pardo. 2014. Reverbalize: A crowdsourced reverberation controller. Paper presented at Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, November 3–7, pp. 739–40.

Seetharaman, Prem, and Bryan Pardo. 2016. Audealize: Crowdsourced audio production tools. *Journal of the Audio Engineering Society* 64: 683–95. [CrossRef]

Tavana, Art. 2015. Democracy of sound: Is garageband good for music? *Pitchfork*, September 30.

Zheng, Taylor, Prem Seetharaman, and Bryan Pardo. 2016. Socialfx: Studying a crowdsourced folksonomy of audio effects terms. Paper presented at 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, October 15–19, pp. 182–86.