NORTHWESTERN UNIVERSITY

Supporting Novice Communication of Audio Concepts for Audio Production Tools

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Mark Cartwright

EVANSTON, ILLINOIS

December 2016

ABSTRACT

Supporting Novice Communication of Audio Concepts for Audio Production Tools

Mark Cartwright

Catalyzed by the invention of magnetic tape recording, audio production has transformed from technical to artistic, and the roles of producer, engineer, composer, and performer have merged for many forms of music. However, while these roles have changed, the way we interact with audio production tools has not and still relies on the conventions established in the 1970s for audio engineers. Users communicate their audio concepts to these complex tools using knobs and sliders that control low-level technical parameters. Musicians currently need technical knowledge of signals in addition to their musical knowledge to make novel music. However, many experienced and casual musicians simply do not have the time or desire to acquire this technical knowledge. While simpler tools (e.g. Apple's *GarageBand*) exist, they are limiting and frustrating to users.

To support these audio-production novices, we must build audio-production tools with affordances for them. We must identify interactions that enable novices to communicate their audio concepts without requiring technical knowledge and develop systems that can understand these interactions.

This dissertation advances our understanding of this problem by investigating three interaction types which are inspired by how novices communicate audio concepts to other people: *language*, *vocal imitation*, and *evaluation*. Because learning from an individual can be time consuming for a user, much of this dissertation focuses on how we can learn general audio concepts offline using crowdsourcing rather than user-specific audio concepts. This work introduces algorithms, frameworks, and software for learning audio concepts via these interactions and investigates the strengths and weaknesses of both the algorithms and the interaction types. These contributions provide a research foundation for a new generation of audio-production tools.

This problem is not limited to audio production tools. Other media production tools—such as software for graphics, image, and video design and editing—are also controlled by low-level technical parameters which require technical knowledge and experience to use effectively. The contributions in this dissertation to learn mappings from descriptive language and feedback to low-level control parameters may also be adapted for creative production tools in these other mediums. The contributions in this dissertation can unlock the creativity trapped in everyone who has the desire to make music and other media but does not have the technical skills required for today's tools.

Acknowledgements

Of course, I'd like to thank Bryan Pardo for being a truly fantastic advisor. Thank you for believing in me and taking on a student without any computer science degrees. Thank you for the invaluable research advice, for truly caring about my future, for all of the exciting discussions, for all of the lunches, and for being a friend during all of the ups and downs of the past seven years. I look forward to future collaborations and lunches (on me).

I'd like to thank the rest of my committee—Darren Gergle and Doug Downey—for taking the time out of your busy schedules for my dissertation and for the career advice in moments of panic.

I'd like to thank my collaborators at Adobe Research and Queen Mary University of London—Gautham Mysore, Matt Hoffman, and Josh Reiss—for inviting me to your institutions, for making time to work with me, and for providing me with new perspectives on research.

I'd like to extend a special thanks to my parents—Tina and Donald Cartwright. Thank you for all of your love and encouragement. You have always been supportive of my musical and computing endeavors—buying me programming books when I was 13, buying me instruments and computers throughout my youth, providing me with the best education you could, and sending me to all of the numerous private music lessons that I begged for—tuba, bass, piano, trombone, tabla. You have also always been supportive of my continued education and the choices that I've made. I love you guys.

I'd also like to thank my sister, my brother-in-law, my nieces—Alison, Craig, Kierstin, and Kelsey Sykes—and all of my extended family. You have always been supportive and provided much comic relief over the phone on my evening commute after hours of staring at my computer screen.

I'd especially like to thank my late grandfather, Dr. Ray Ameen. While you passed away only months before I applied for this PhD, you were always extremely supportive and encouraging of everything I did. Your talent, intellect, integrity, and kindness is unmatched. You continue to be an inspiration and role model.

I'd like to thank my friends and bandmates in volcano!—Sam Scranton and Aaron With. You are both amazing friends. Not only have you kept me excited about music over the years, but our interactions actually inspired the topic and approach of my dissertation. Thanks!

I'd like to thank my current and former labmates and colleagues at the Interactive Audio Lab—Prem Seetharaman, Bonjun Kim, Fatemeh Pishdadian, Ethan Manilow, Zafar Rafii, Zhiyao Duan, Jinyu Han, David Little, Andy Sabin, Denis Lebel, and Arefin Huq. Thanks for the fruitful discussion about research, audio, music, and life. It would have been a boring seven years without you all—you are the best procrastination. Special thanks to Arefin Huq for the computer-science kung-fu training in my first and longest year.

I'd like to thank my other computer-science colleagues, officemates, and cohort—Maria Chang, Michael Lucas, Matt McLure, Yi Yang, Madhav Suresh, Joe Blass, Thanapon

Noraset, Dave Demeter, Chen Liang, Dev Ghosh, Patrick McNally, Zeina Leong, Chris Heinrichs, Siddharth Sigtia, Brecht De Man, Anders Øland, Jon Ford, and Aaron Karp. Our discussions have broadened my knowledge. I'd especially like to thank Maria Chang for all the moral support over the years.

I'd also like to thank the HCI group at Northwestern—Emily Harburg, Scott Cambo, Robin Brewer, Sara D'Angelo, Julie Hui, Noah Liebman, Jordan Davison, Darren Gergle, Anne Marie Piper, Haoqi Zhang, Elizabeth Gerber, Jeremy Birnholtz, and Michael Horn—for accepting me into your community and always giving me great advice.

I'd of course like to thank all of my non-academic friends in Chicago and around the world for all the support and good times—James Wood, David Norelid, Tia Hansen, Mike Gelety, Joe Tepperman, Kern Saxton, Victoria Norelid, Luke Dahl, Julia Wood, Alastair Wood, Emma Moore, Matt Dillon, Adam Kader, Jenny Maoloni, Molly Scranton, Neha Bhardwaj, Chris Sherman, Lindsay Moore, Nick Siegel, Heather Mingo, John Sutton, Cassie Meyer, Jenny Abrahamian, Michael Neuner, Anna Mormolstein, Hannah Gamble, Melissa Eubanks, Pete Snyder, Lindsay Bosch, Marla Campbell, and Nell Haynes and the Haynes family.

I'd like to thank my undergraduate and masters advisors and professors—Gary Kendall, Julius O. Smith III, Virgil Moorefield, and Jonathan Abel—for encouraging and inspiring me long ago.

I'd like to thank the National Science Foundation. I couldn't have done any of this research without their funding from the Graduate Research Fellowship DGE-0824162 and Grants IIS-0757544, IIS-0812314, IIS-1116384, and CHS-1420971.

Lastly, I'd like to thank Kristina Francisco for supporting me with her love, patience, and understanding; for not minding the late nights and early mornings; for all of the meals that I was too busy to cook; and of course for following me around the country to pursue dreams and adventures.

List of Abbreviations

AMT: Amazon's Mechanical Turk 73, 85, 91, 99, 100, 102, 104, 107, 108, 110, 112, 122, 123, 137, 139, 143, 152, 186, 187, 192–194, 197, *Glossary:* Amazon's Mechanical Turk

ANOVA: analysis of variance 118, 142

CAQE: Crowdsourced Audio Quality Evaluation 19, 20, 103, 120, 152, 153, *Glossary:* Crowdsourced Audio Quality Evaluation

CI: confidence interval 19–23, 74, 113, 117, 118, 140–143, 146–148, 181, 182

DAW: Digital Audio Workstation 16, 24, 25, 27, 28, 35, 37, 41, 90, *Glossary:* Digital Audio Workstation

DMI: digital musical instrument 163, 164

DNN: deep neural network 202

DTW: dynamic time warping 174, 178, 203

EQ: equalization 66, 69, 154, *Glossary:* equalization

ERB: equivalent recatangular bandwidth 18, 69, 72, 76, 77, *Glossary:* equivalent recatangular bandwidth

FM: frequency modulation 33, 168

HCI: human-computer interaction 40

HIT: Human Intelligence Task 85, 107, 108, 122, 123, 144, 193, *Glossary:* Human Intelligence Task

HSD: honest significant difference 118, 143

ISR: image-to-spatial-distortion ratio 105, 114, 139, *Glossary:* image-to-spatial-distortion ratio

ITU: International Telecommunication Union 98

KL divergence: Kullback-Leibler divergence 18, 80, 203, *Glossary:* Kullback-Leibler divergence

MCMC: Markov chain Monte Carlo 129, 135

MDS: multi-dimensional scaling 18, 80, 83, 154, *Glossary:* multi-dimensional scaling

MFCC: mel-frequency cepstral coefficient 203

MUSHRA: Multi Stimulus test with Hidden Reference and Anchor 15, 19–21, 97–104, 106–109, 112, 113, 115–119, 122, 128, 140–142, 144, 146–151, 155, *Glossary:* Multi Stimulus test with Hidden Reference and Anchor

NUTS: No-U-Turns Sampler [**73**] 135

OED: Oxford English Dictionary 83, 84

PEASS: Perceptual Evaluation methods for Audio Source Separation 99, 101, 104–107, 109, 112, 152, *Glossary:* Perceptual Evaluation methods for Audio Source Separation

PSA: prioritized shape averaging 178

QBE: query-by-example 162, 169, 173

QBH: query-by-humming 54, 163, *Glossary:* query-by-humming

RMS: root mean square 115

RSC: relative spectral curve 18, 72, 76, 77, 79, 82, *Glossary:* relative spectral curve

SAE: stacked auto-encoder 202

SAR: sources-to-artifacts ratio 105, 114, 139, *Glossary:* sources-to-artifacts ratio

SDR: source-to-distortion ratio 105, 114, 139, *Glossary:* source-to-distortion ratio

SIR: source-to-interference ratio 105, 114, 139, *Glossary:* source-to-interference ratio

SISEC: Signal Separation Evaluation Campaign 105, *Glossary:* Signal Separation Evaluation Campaign

SVM: support vector machine 202

TSR: transitivity satisfaction rate 135

WST: weak stochastic transitivity 22, 135, 145, 154

Glossary

affordance: As defined by Don Norman, "...the term affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used" [**127**] 17, 38, 40, 41, 44, 45, 49–52, 58, 92, 94, 95, 156

Amazon's Mechanical Turk: A microtask labor market in which workers complete Human Intelligence Tasks for monetary rewards 73, 99, 104, 107, 122, 152, 186

anchor: A stimulus in a subjective evaluation test that has an expected rating. For example, a very poor quality audio example that would be expected to be rated poorly on an overall quality scale 19–21, 102, 104, 106, 109, 111–113, 117, 122, 127–129, 139, 140, 142, 148, 150, 151

audio concept: An idea of sound, type of sound, or "character" of a sound 17, 38, 46, 47, 49–54, 56–58, 67, 72, 91, 92, 94–96, 154, 156–159, 161, 162, 178, 183–186, 188, 200, 202, 203, 205

audio production: The manipulation and design of audio for consumer media. It is a general term that encompasses audio manipulation/design tasks in music production, sound recording, audio post-production, audio mixing, mastering, live sound, and sound design (for music, film, theater, games) 25, 28, 34, 35, 39, 41–46, 48–51, 53, 56, 58, 59, 63, 67, 88, 89, 92, 94, 95, 156, 157, 186

audio source separation: A task that aims to extract a clean recording of a one or more target sounds (e.g. a vocalist or a choir of vocalists) from a recording containing a mixture of several sounds (e.g. the rest of a band) [**195**] 98, 99, 101, 104–106, 109, 116, 151

BSS-Eval: A set of automatic objective measures (SDR, ISR, SIR, SAR) for audio source separation developed by Vincent et al [**193, 194**] that consist of energy ratios of source signals to projections of source signals onto other signals 19–21, 99, 101, 104, 105, 113, 114, 116, 139–141

channel strip: A set of audio processing modules that are associated with a single audio channel on a mixing console. It typical includes processing modules such as a microphone preamplifier, output gain, and equalization. They may also include processors such as compressors, noise-gates, limiters, etc 41, 42

Crowdsourced Audio Quality Evaluation: A software toolkit that I developed that enables researchers to quickly and easily evaluate audio quality using crowdsourcing platforms 19, 103

descriptive term (also **descriptor**)**:** A word describing audio (e.g., *warm*). Also referred to as a *descriptor* for brevity 14, 17, 18, 56, 58, 61–70, 72, 74, 75, 77–81, 83, 85, 88, 90, 91, 153, 154

descriptor: *See* descriptive term 14, 18, 51, 58, 62, 64–67, 69, 72, 74–78, 82, 83, 85, 87, 88, 90–92, 154

descriptor definition: In SocialEQ, this is a general audio concept comprised of a set of personal audio concepts that all represent the same descriptive term. A definition may be vague or precise depending on how much agreement there is between personal audio concepts that share the descriptive term 14, 18, 66, 79, 81–83, 86, 94, 153

Digital Audio Workstation: An electronic device or software application for recording, editing, mixing, and processing audio 16, 24, 90

equalization: An audio process which adjusts the frequency balance of the signal using filters 14, 17, 18, 56, 59, 61, 62, 65, 66, 68, 69, 72, 75, 77, 78, 80, 87, 88, 90, 154

equalizer: An audio processing tool that performs equalization 17, 56, 58–63, 65–67, 72, 77, 78, 88, 90

equivalent recatangular bandwidth: A psychoacoustical measure that gives an approximate bandwidth of human hearing filters given a center frequency [**123**] 69

Human Intelligence Task: A unit of work on Amazon's Mechanical Turk 85, 107, 193

image-to-spatial-distortion ratio: An audio source separation evaluation measure in BSS-Eval 105

Kullback-Leibler divergence: A measure for calculating the distance between two probability distributions [**72**] 82, 203

Multi Stimulus test with Hidden Reference and Anchor: ITU Recommendation ITU-R BS.1534-2 [**82**], also known as MUSHRA, is a listening tests that consists of rating up to 12 audio stimuli on a scale from 0-100 in comparison to a reference stimulus. Of the stimuli that are evaluated, one is a hidden (i.e. unlabeled) reference, at least one is a hidden low anchor (e.g. a stimulus that is expected to be rated low), and possibly one or more medium anchors (e.g. stimuli that are expected to be rated medium) 15, 97

multi-dimensional scaling: An algorithm to place objects in a N-dimensional space based on the distance between the objects [**13**] 18, 80, 154

Perceptual Evaluation methods for Audio Source Separation: A set of automatic objective evaluation measures for audio source separation developed by Emiya et al [**42**] that utilizes auditory models and are trained on human subjective evaluation data 99

personal audio concept: In SocialEQ, this is an audio concept learned in a single session consisting one person's interpretation of a descriptive term (e.g., the session where Bob teaches 'warm' to SocialEQ). It is represented by a relative spectral curve 66, 72, 74, 78, 79, 81–83, 94

query-by-humming: The process of searching for a piece of music by singing or humming a portion of it [**135**] 54, 163

relative spectral curve: A set of relative gains (i.e., boosts or cuts) on the 40 ERB frequency bands. It is used to represent a personal audio concept in SocialEQ 17, 18, 69–72, 153, 154

Signal Separation Evaluation Campaign: An annual evaluation of audio source separation algorithms [**192**] 105

sonic interaction design: "Sonic interaction design is positioned in between auditory display, interaction design, ubiquitous computing and interactive arts. In SID, the role of sound as natural carrier of information is exploited as effective means to establish continuous negotiation with interactive artifacts." [**121**] 200

source-to-distortion ratio: An audio source separation evaluation measure in BSS-Eval 105

source-to-interference ratio: An audio source separation evaluation measure in BSS-Eval 105

sources-to-artifacts ratio: An audio source separation evaluation measure in BSS-Eval 105

synthesizer: An electronic musical instrument that generates audio signals that replicate acoustic instruments or are abstract in nature 16, 22, 24, 26, 27, 31–33, 36–42, 45, 62, 90, 157, 161

timbre: "That multidimensional attribute of auditory sensation which enables a listener to judge that two non-identical sounds, similarly presented and having the same loudness, pitch, spatial location, and duration, are dissimilar" [**2**] 33, 45, 54, 64, 65, 77, 158, 163, 164, 167

track: The audio signal channel on a storage device. For example, a four-track audio recorder can recorded four separate audio signals that are stored on their own tracks and can be played back and manipulated independently of each other 24, 25, 31, 34, 42

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

> 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 lot of ideas. (Anonymous [**3**])

Imagine you're the author of the quote above. You've been playing guitar for 30 years and you want to learn how to record and produce pop music. So, you go out and buy a recording interface and a Digital Audio Workstation (DAW) application (e.g. Apple's Logic). First you figure out how to record your guitar. After you manage this step, you then decide to fill out your track with a synth line—you have a distant, warbly sound in mind. So, you open up Apple's ES2 synthesizer (see Figure 1.1), a typical digital synthesizer, and suddenly you're lost. The synthesizer interface is littered with knobs, sliders, and drop-down boxes that control 125 low-level technical parameters. What do "LFO1 Rate" and "CBD" mean? (see Figure 1.2) Without having this technical knowledge, you can twiddle some knobs around and may be find something interesting... but you are unlikely going to find that distant, warbly sound that you have in mind. Even if each of these parameters could only take two values, there would still be 2^{125} parameter combinations... it's doubtful you'll find your desired sound by simply wandering around such a *huge* parameter space. You then notice there are presets that specify saved parameter settings. You start browsing these presets, but they are all named completely uninformative names like “Glitterworm” or “Soft-cutter” (see Figure 1.3). Skipping this first dilemma for now, let’s say that you are able to find some sounds that you like and you record a few lines. When you play the lines back together, it sounds terrible—it’s a muddy mess. To fix this, you have to process and combine the tracks together to form a coherent mix. Where do you begin? A typical DAW (see Figure 1.4) may have 80 different audio processing modules, each of which has several or dozens of parameters (see Figure 1.5). Which modules can fix your mix? Which recorded lines do you need to process? How do you choose the parameters? It’s overwhelming... you give up.

Why are these tools designed like this? How can we design audio production tools that don’t require years of experience and technical knowledge? How can we enable users who would have otherwise given up to express their creative ideas?

I seek the answers to these questions. **To build audio production tools for people without years of experience and technical knowledge, I look to the methods with which people communicate sound ideas to each other. I then develop methods for software to understand these communication methods so that people can communicate their ideas to software with ease and to focus on creating.**

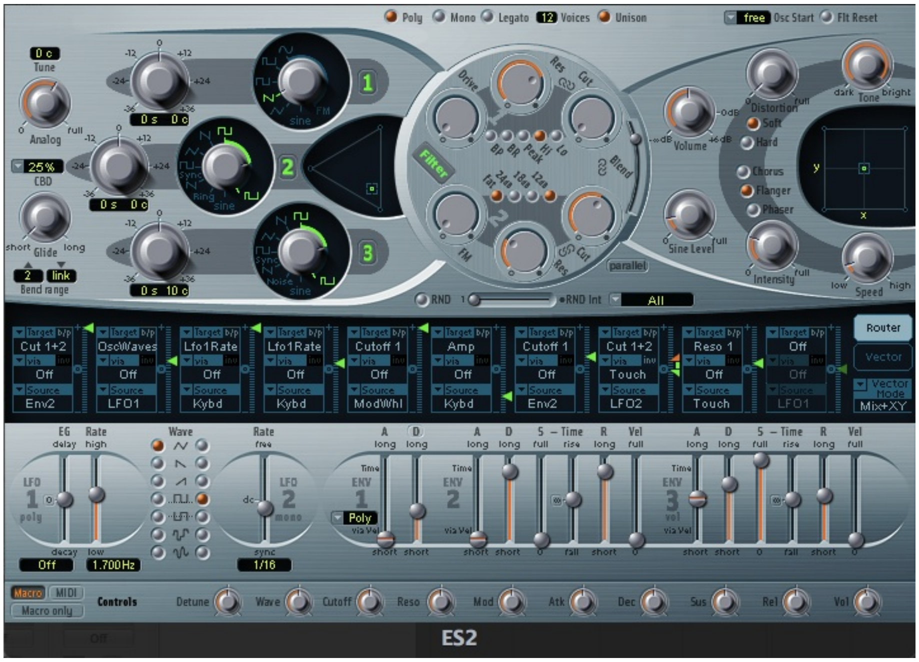


Figure 1.1. Screenshot of Apple's ES2, a typical software synthesizer. Introduced in 2001, it is still shipped with Apple's Logic Suite.

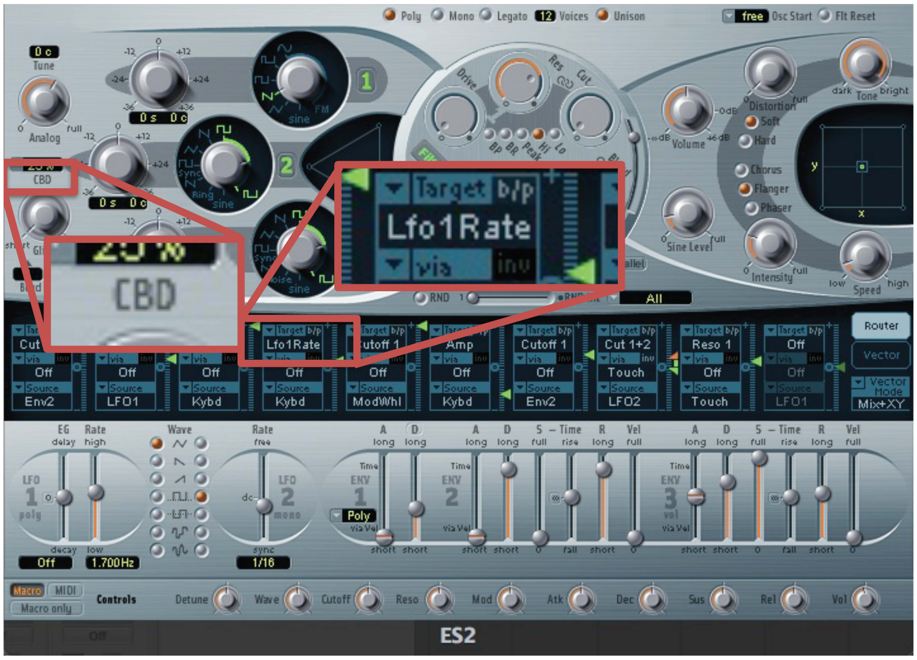


Figure 1.2. "LFO" and "CBD" are parameters in Apple's ES2 synthesizer.

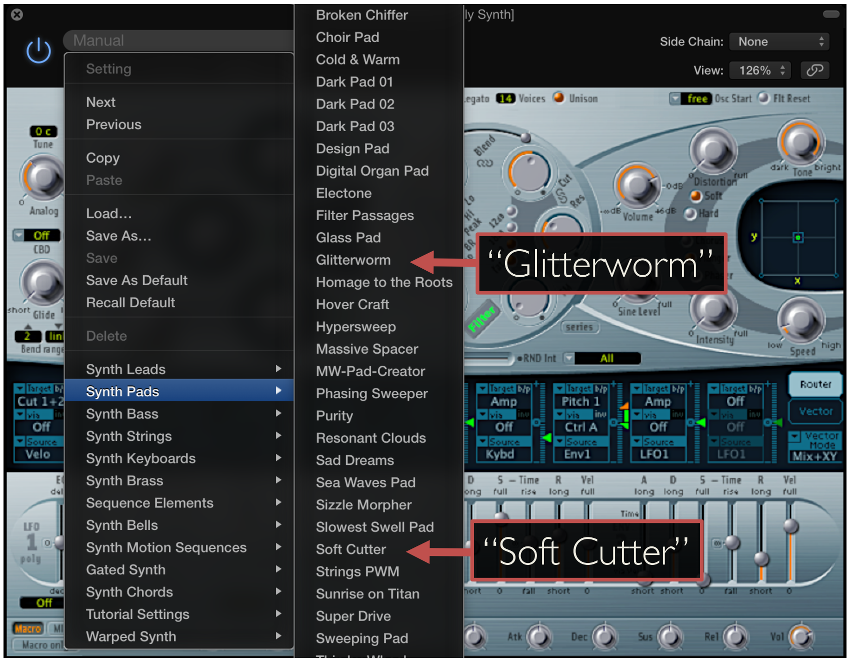


Figure 1.3. Presets in Apple's ES2 synthesizer.

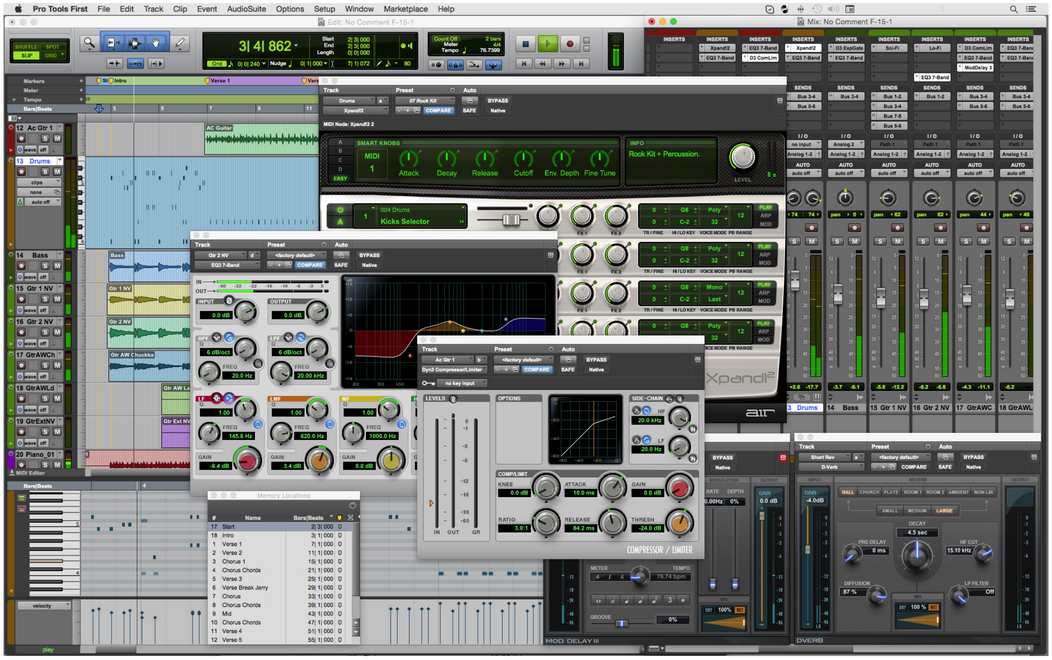


Figure 1.4. A screenshot of Avid's Pro Tools, a typical DAW. Source: `http://www.avid.com` (downloaded February 2016)

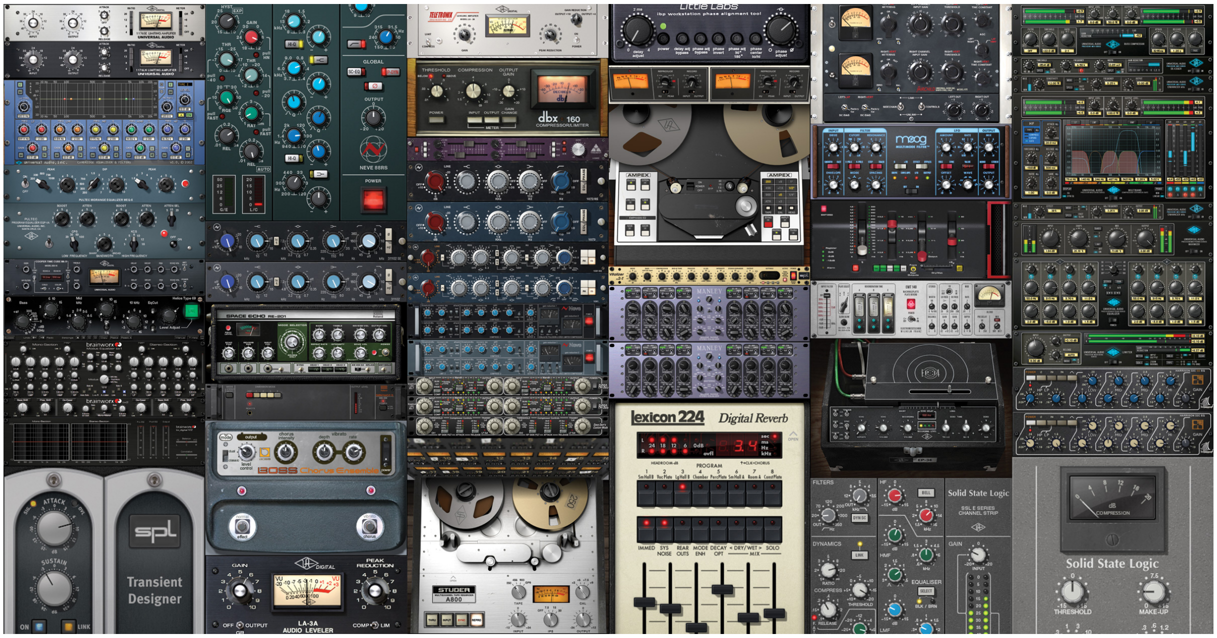


Figure 1.5. A suite of DAW plugins (audio processing modules) by Universal Audio. Source: `http://www.uaudio.com` (downloaded February 2016)

1.1. Background and Motivation

Audio production is the manipulation and design of audio for consumer media. It is a general term that encompasses audio manipulation/design tasks in music production, sound recording, audio post-production, audio mixing, mastering, live sound, and sound design (for music, film, theater, games). However, in this dissertation, I focus on use cases of audio production by music producers (i.e. music production).

In this section, I first explain how audio production technologies became an integral part of music-making in the last century. I then explain that while personal computers and digital audio software have made powerful audio tools economically affordable, digital audio tools still rely on interaction metaphors established by analog hardware in the middle of the 20th century and lack interaction affordances for novices unfamiliar with these analog counterparts. Therefore, the power and flexibility of these complex audio tools remains inaccessible to users without experience or the time to learn them.

1.1.1. The Evolution of the Role of Audio Production in Music

Before sound-recording technology, music was ephemeral, tied to a specific time and space. Music was documented through written scores and could only be heard when a musician interpreted and performed the notes and markings on a score. After a musical performance ended, the music vanished. You could never hear the exact same piece of music twice. Audio recording changed this. As Brian Eno said, once you can record music, "you're in a position of being able to listen again and again to a performance, to become familiar with details you most certainly had missed the first time through, and to become very fond of details that weren't intended by the composer or the musicians. The effect of this on the composer is that he can think in terms of supplying material that would actually be too subtle for a first listening" [**43**].

The first sound recording and playback medium was the Edison's phonograph cylinder [**15**]. Introduced in 1877, a phonograph transduced acoustic signals to mechanical vibrations which were impressed into wax cylinders—and later wax discs [**15**]. Twenty years later, Valdemar Poulsen patented the telegraphone, which recorded sound magnetically onto steel wire [**15**]. During World War II, German engineers invented the AC-biased magnetic tape recorder (the magnetophon), which Jack Mullin, a U.S. Army engineer, brought back to the United States after the war [**15**]. With the help of Ampex, it became a commercial success. [**15**]

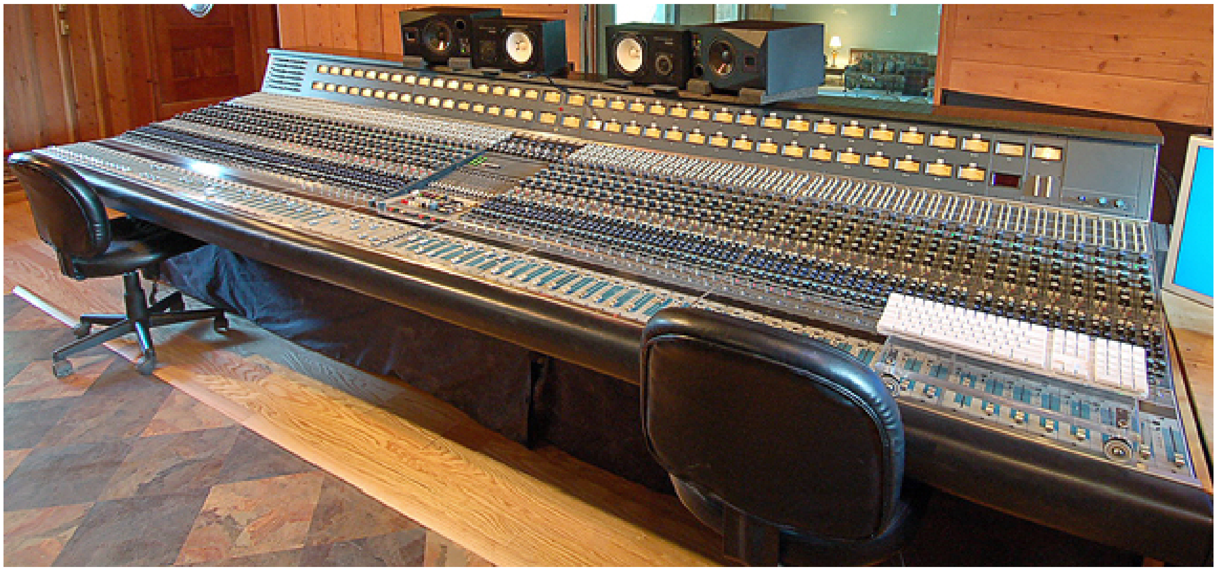


Figure 1.6. A Neve 8078, a typical mixing console in the 1970s. Source: `https://commons.wikimedia.org/wiki/File:SonicRanchNeveConsole.jpg` (downloaded June 2016)

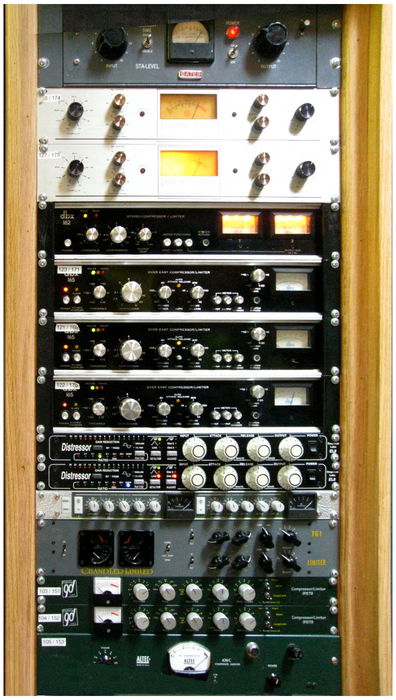


Figure 1.7. An equipment rack full of audio processing modules. Source: `https://en.wikipedia.org/wiki/File:Comp._rack_(Supernatural).jpg` (downloaded June 2016)

However, it wasn't just the dramatically improved sound quality that made magnetic tape such a successful recording medium—it was also the first practical editable recording medium [**15**]. You could splice tape and rejoin it using adhesive tape or glue. Prior to tape, recordings were "direct-to-disc", requiring all performers to play with absolute precision since practical editing was not possible. With tape however, recordings became "malleable and mutable" [**43**]. Though editable, the early days of tape recording still required all performers to play at the same time. This changed when Ampex released the multi-track tape recorder that allowed instruments to be recorded on their own tracks and manipulated in isolation after recording. Two-track recorders were common in the 1950s, and three and four-track recorders were common in the 1960s [**15**]. However, this number quickly increased. By the early 1980s, recording studios were syncing together multiple 24-track tape machines to obtain up to 72 tracks [**15**].

As the number of tracks increased, so did the complexity of recording studios, adding numerous new audio processing tools to take advantage of this new flexibility. These tools included equalizers, filters, reverberators, delays, compressors, limiters, expanders, noise gates, multi-band compressors, dynamic equalizers, overdrive, distortion, exciters, chorus, flanger, phasers, pitch/time processors and more. Even more tools have been introduced since the advent of commercial digital audio tools, including harmonizers, pitch correction (e.g. auto-tune), spectral noise reduction, and more. All of these tools are manipulated using knobs and sliders that control low-level technical parameters. These knobs and sliders fill giant consoles and walls of equipment racks (see Figure 1.7) in recording studios.

The 1960s and 1970s, also saw the introduction of commercial analog and digital synthesizers. While inventors developed other electronic instruments earlier in the 20th

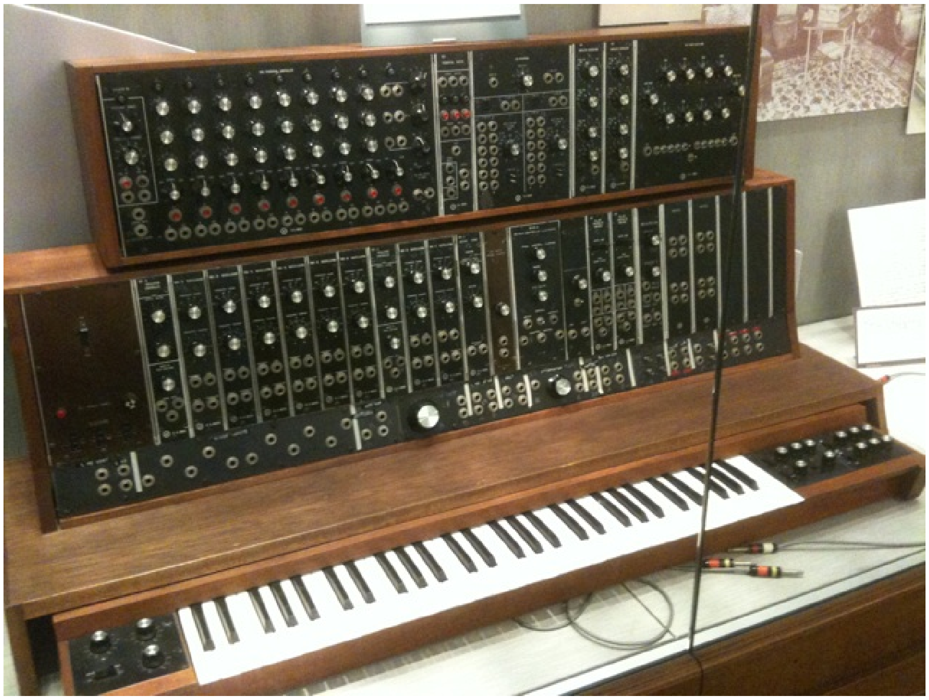


Figure 1.8. The first commercially sold Moog synthesizer (1964). Source: `https://en.wikipedia.org/wiki/Moog_modular_synthesizer` (downloaded June 2016)

century, as Théberge notes, "the vast majority of these early devices were, from a practical point of view, probably poorly designed in the first place: idiosyncratic, incapable of functioning in any musical context outside the laboratory, or impossible to manufacture in a cost-efficient manner" [**185**, 43]. Instruments such as the theremin and the ondes martenot achieved some success with a few well-known performers and composers, but even these instruments failed to be adopted by a wider audience at the time[**185**, 44]. Electronic instruments were not a commercial success until inventors and entrepreneurs such as Bob Moog and Donald Buchla started making commercial modular synthesizers in the 1960s (see Figure 1.8) and then smaller, electronic keyboards such as the extremely popular *Minimoog* in the 1970s [**185**, 52]. These first commercial synthesizers utilized subtractive synthesis, but later manufacturers introduced many other types of synthesis:

additive, sampled-based, digital frequency modulation (FM), granular, and more. While the type of synthesis may vary, the sound of all of these synthesizers is typically programmed with the same interaction paradigm of the audio processing tools: by using the knobs and sliders which control low-level technical parameters.

Tape splicing and multi-track recording revolutionized sound recording, but as Jones notes "recording technology did more than liberate music from the constraints of space and time. Along with electronic instruments, it enabled music to be organized around sound and timbre instead of notes [**90**, 48]. This organizational shift from notes to timbre was proposed by Italian Futurust Luigi Russolo in his famous 1913 manifesto, "The Art of Noises" [**155**], and reiterated by composers such as Edgard Varse (1936) [**190**] and John Cage (1937) [**17**]. However, it wasn't until the advent of tape-splicing, multi-track recording, and commercial electronic instruments that this shift began occurring in mainstream music and popular musicians began breaking away from the constraints of physical acoustics.

As noted by numerous scholars and artists, this shift propelled recording from a primarily technical task to an artistic one [**124**]. Composer/Scholar/Producer Virgil Moorefield says this shift changed recording's metaphor from "from one of the 'illusion of reality' (mimetic space) to the 'reality of illusion' (a virtual world in which everything is possible)" [**124**]. Composer/Producer Brian Eno said "In a compositional sense this takes the making of music away from any traditional way that composers worked, as far as I'm concerned, and one becomes empirical in a way that the classical composer never was. You're working directly with sound, and there's no transmission loss between you and the sound - you handle it." [**43**]. Producer André Allen Ajos stated "The mixing process is part of the songwriting process and effects should be treated like songwriting tools" [**153**]. Because of this shift, audio production is a vital part of contemporary music, and it is common for the roles of the record producer, composer, recording engineer, and performer to have all merged [**124**, 112].

While the influence on audio technology on the sound of popular music is very apparent, it has also transformed the way that even "traditional" forms of music are recorded. In the first quarter of the twentieth century for instance, a jazz ensemble was recorded in one take with well-placed musicians and a single megaphone-like recording horn in a room [**15**]. In the second quarter of the twentieth century with introduction of electrical recording, there was often more than one microphone, but the recording was still one-take and direct-to-disc [**15**]. However, when multi-track recording was introduced, this process changed—every drum and instrument now has a microphone assigned to it, and each is recorded to a separate track. Solos are edited from several takes to perfection, and all of these musical fragments are processed and put back together so that it sounds like you are listening to a perfect performance from the perfect vantage point. The aesthetic is to make the technology as transparent as possible, as if you are there—this audio production aesthetic is like the audio equivalent of documentary photography whereas popular music's audio production aesthetic is more like fashion photography. However, the "perfect vantage point" of this "documentary"-style production often could never exist in reality—it's more like a hyper-reality. But strangely, after listening to recordings like this for decades, this constructed hyper-reality is what we are now most familiar with and what we expect. For example, the "O Fortuna" from Carl Orff's *Carmina Burana*

(1937) that we are familiar with from countless movies and trailers is not the same piece that Orff originally wrote. The role of audio production has changed the music.

1.1.2. The "Democratization" of Audio Production Tools

1.1.2.1. The rise of the digital home studio. In 1989, the audio quality of professional recording equipment began making its way into home studios—Digidesign released SoundTools, the first commercial, random-access DAW, and in 1991, they introduced ProTools, which is still the industry-standard DAW 25 years later [**15**, 144]. DAWs are digital audio recording solutions with PC software front-ends that allow nondestructive, random access, stereo, audio editing. While they initially only had simple digital signal processing functions, eventually DAWs moved all of the audio production capabilities of professional recording studios to the personal computer, and the costs of the software and required hardware dropped dramatically during the 1990s. Like the introduction of multitrack tape recorders, this shift also revolutionized music. It enabled technically-skilled musicians to produce professional sounding music from the comfort of their home and at a fraction of the cost. With a personal computer, a recording interface, and a DAW application, musicians now have unlimited access to tools similar to those that previously were only accessible by renting a recording studio for hundreds of dollars an hour.

With the introduction of DAWs, experienced users wanted to emulate *virtually everything* about hardware-based studios in software. Software was more cost-efficient, more flexible, and required less maintenance. However, users wanted to get the same characteristic sounds out of software tools as they did with their hardware counterparts. For example, while all analog dynamics compressors reduce the dynamic

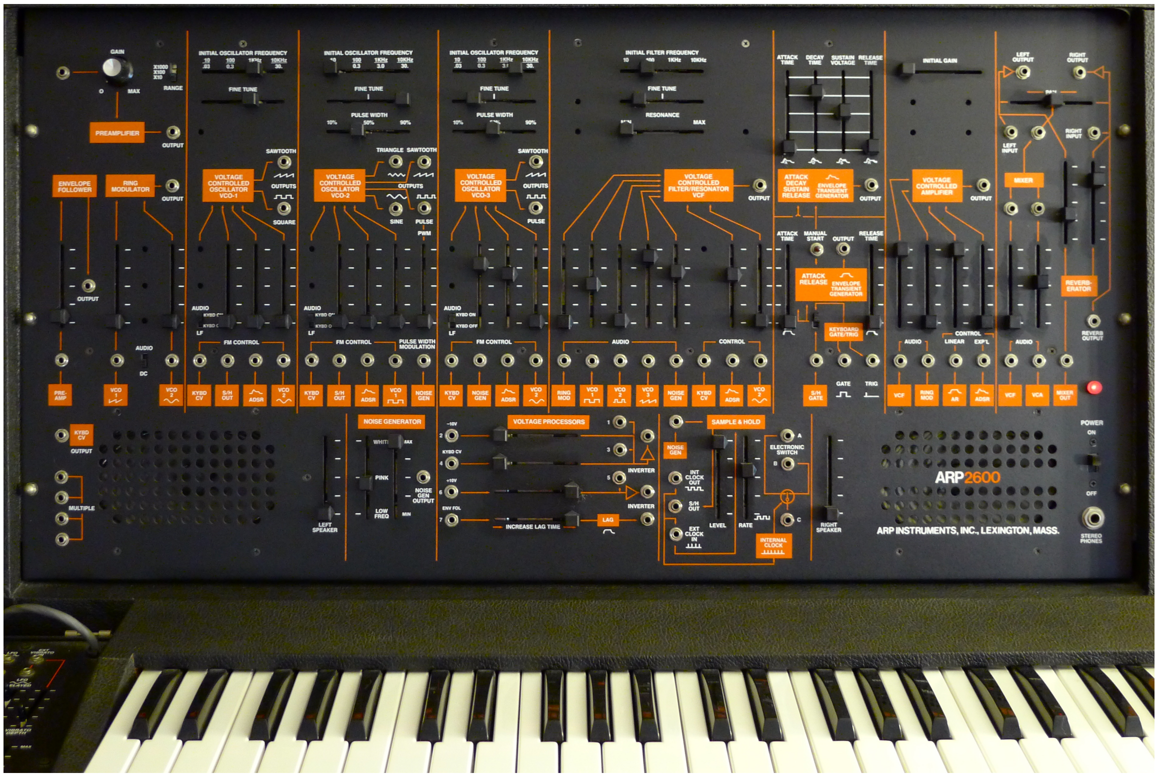

PREAMPLIFIER
ENVELOPE FOLLOWER
RING MODULATOR
VOLTAGE CONTROLLED OSCILLATOR VCO-1
VOLTAGE CONTROLLED OSCILLATOR VCO-2
VOLTAGE CONTROLLED OSCILLATOR VCO-3
VOLTAGE CONTROLLED FILTER/RESONATOR VCF
ATTACK DECAY SUSTAIN RELEASE
ENVELOPE TRANSIENT GENERATOR
VOLTAGE CONTROLLED AMPLIFIER
MIXER
REVERBERATOR
NOISE GENERATOR
VOLTAGE PROCESSORS
SAMPLE & HOLD
ELECTRONIC SWITCH
INTERNAL CLOCK
ARP2600
ARP INSTRUMENTS, INC., LEXINGTON, MASS.

Figure 1.9. The ARP 2600 synthesizer (1971 - 1981). Source: `http://citizenfitz.com/logs-2011-06-02-ARP-2600-synthesizer/arp-2600-synth-large-picture.jpg` (downloaded February 2016)

range of an audio signal, they do so using slightly different circuits that use different electronic components. Some tools impart characteristics onto the audio signal that have become highly desired. Therefore, much research and development has gone into modeling expensive, sought-after hardware tools in software. The developers of these software tools didn't stop with just emulating the sound though, they also emulated the look and feel of hardware tools in the skeuomorphic graphical user interfaces of the new software tools (see Figure 1.9 and 1.10). Even software tools that aren't directly modeling existing hardware tools have adopted the "hardware look and feel"—Propellerhead's Reason fully embraces skeuomorphism, adopting graphics of equipment racks, mixing consoles, screws,

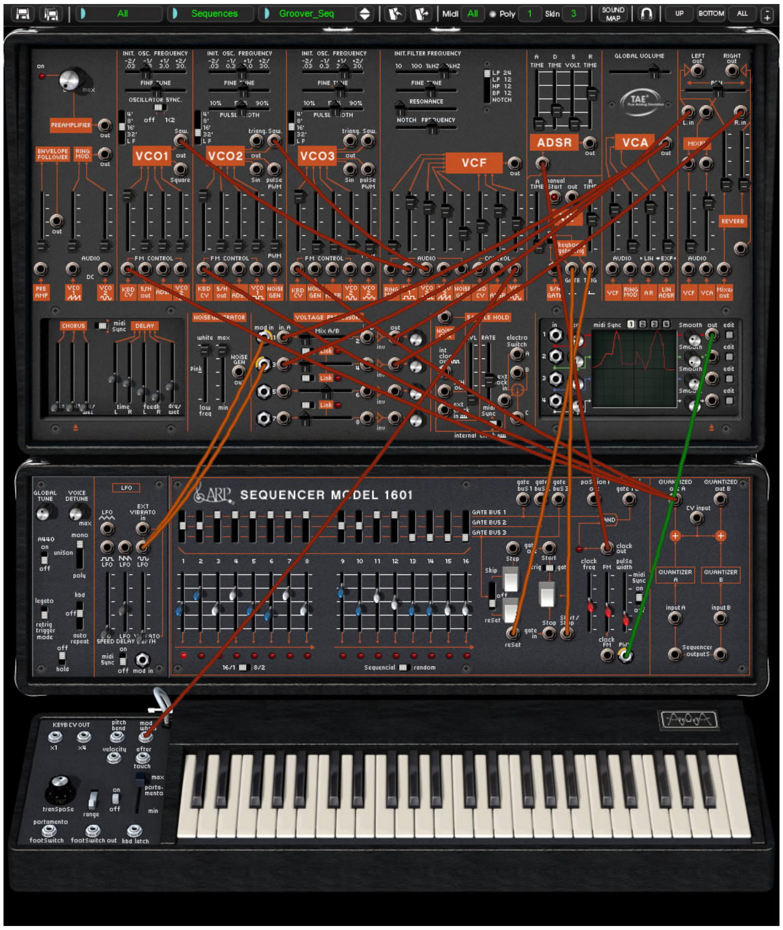


Figure 1.10. Arturia's ARP2600V, a software emulation of the ARP 2600 synthesizer (2005 - present). Source: `https://www.arturia.com/products/analog-classics/arp2600v` (downloaded February 2016)

LED-style displays, and audio cables that appear to obey the laws of gravity 1.11. **Skeuomorphism is the norm for commercial digital audio processing and synthesis software.** Therefore, audio processing and synthesis tools in DAWs use the same interaction paradigm used in analog hardware devices: knobs and sliders that control low-level technical parameters. Though with software DAWs, users control the knobs and sliders by using a mouse to click on their respective screen widgets. Because professional users were familiar with the interaction conventions used in traditional hardware-based studios,

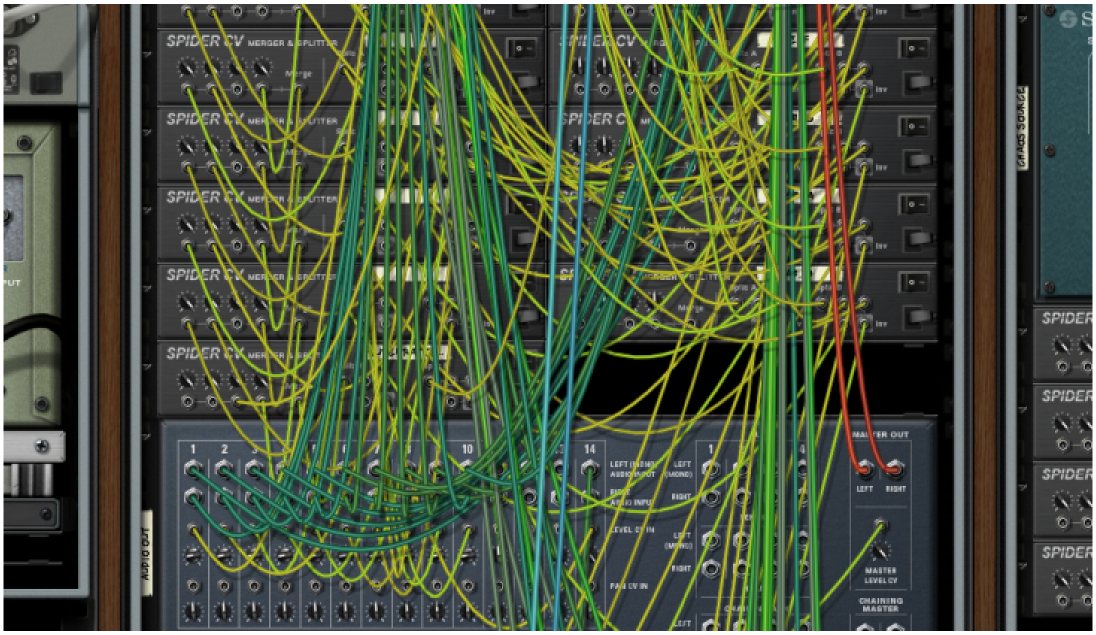


Figure 1.11. A screenshot of Propellerhead's Reason. Source: `http://rackextensionreview.com/cv-reason-behind-front-panels/` (downloaded July 2016)

retaining the same interaction conventions eased the transition from hardware-based to software-based studios [**8**].

However, new users of audio processing tools have likely never used hardware devices such as analog mixing consoles or analog synthesizers—they are likely unfamiliar with the interaction metaphors of hardware tools. Audio software tools have affordances that may be perceived by experienced audio engineers but not by inexperienced users. **It takes considerable technical and theoretical knowledge to effectively communicate a desired audio concept to software via low-level technical parameters** [**185**, 211]**.** The time it takes to learn this knowledge is similar to learning a new instrument. Some new users may embrace this challenge, enjoying spending their time twiddling knobs and reading books and manuals until they learn the tools. **However, many users simply want to obtain their desired sound and move on.**

1.1.2.2. Commercial attempts to support novice users. To support these users, **manufacturers of audio production tools have attempted to address the knowledge gap of users with presets, simplified interfaces, and pre-fabricated sounds.** Synthesizer patch storage/recall and presets were first introduced in the late 1970s in Sequential Circuit's Prophet-5. The Prophet-5 came with 120 factory preset synthesizer settings. At first, manufacturers believed that users didn't want prefabricated sounds [**185**, 75]. They assumed that all synthesizer players were also programmers that created their own sounds, and the patch storage was simply for users to save their own creations. **However, by the end of the 1980s, manufacturers estimated that only 10 percent of users programmed their own sounds, and the rest of users relied on presets due to the overwhelming complexity of programming** [**185**, 75].

Synthesizer presets enable non-technical users to explore a small sampling of the synthesis space without programming, but they also of course limit the range of sounds a user can produce. This limited range sounds can homegenize the sound of music. According to Théberge, "musicians complained that the limited range of sounds built into some drum machines and synthesizers virtually forced them to write music in a particular style" [**185**, 1]. In fact, Rolling Stone magazine once called the late 1970s and early 1980s the era of "push-button rock" [**106**] due to musicians' reliance on built-in synthesizer and drum sounds.

One way synthesizer manufacturers have addressed this problem is to simply make more presets. Native Instruments' *Komplete* suite currently comes with 17,000 sounds. **However with so many presets, the problem then becomes searching through**

presets instead of directly searching the parameter space. The often uninformative names of presets make this problem even worse. For example, of the 120 presets in the Prophet-5, some of them were named according to acoustic instruments of which they were reminiscent (e.g. "harpsichord" and "flutes"), but others were simply given evocative names (e.g. "final frontier" and "the landing") rather than informative, descriptive names. When searching for a particular sound, preset names such as "final frontier" and "the landing" are not helpful. This trend of uninformative synthesizer preset names still continues (see Figure 1.3). Its continued existence is likely because abstract sounds can be very difficult to describe with language [**102**]. As Lemaitre noted in [**102**] and will be discussed and addressed later in this dissertation, there are times when language can be effective for communicating audio concepts (see Chapter 2), but for very abstract sounds, language often fails to effectively communicate the audio concept (see Chapter 4).

With respect to human-computer interaction (HCI) literature, a preset is usable (i.e. a user can select it), but depending on the preset's label and the user's knowledge and experience, it may not be useful [**115**]. Gibson coined the term *affordances* in his book *The Ecological Approach to Visual Perception* [**56**]: "...the affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill" [**56**, 127]. In other words, he defines an affordance as an action capability of an individual in an environment. In his view, an affordance exists regardless if the individual can perceive it [**115**]. Norman co-opted this term for the design of objects upon which actions can be performed. In Norman's definition, "...the term *affordance* refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used. A chair affords ('is for') support and, therefore, affords sitting. A chair can also be carried" [**127**]. Norman considers an affordance to be a property that is perceived but may not exist, that suggests how to use the property, that can be dependent on the experience/knowledge/culture of the individual, and that can make an action easy or difficult [**115**]. According to McGrenere and Ho, Norman confounds *usefulness* and *usability* in his definition [**115**]. They claim that "The usefulness of a design is determined by what the design affords (that is, the possibilities for action in the design) and whether these affordances match the goals of the user and allow the necessary work to be accomplished. The usability of a design can be enhanced by clearly designing the perceptual information that specifies these affordances" [**115**]. Therefore, while a preset is a perceived affordance, it may not be a useful perceived affordance since it may not match the goals of the user as conveyed by its often uninformative label.

While many DAWs have other audio production tools in addition to synthesizers that utilize presets (e.g. equalizers, compressors, reverberators, etc.), Apple's *GarageBand* (2004-present) fully embraces the preset approach to audio production. GarageBand is Apple Inc.'s DAW that is included for free on all MacOS and iOS devices and is marketed for virtually any user of those devices. It is the commercial benchmark for beginner-friendly DAWs. The latest GarageBand (v10.1.0) includes the typical synthesizer and audio processor presets, but it also includes audio-track "channel strip" presets, instrument-track "channel strip" presets, automated-drummer presets, and project presets. Audio-track "channel strip" presets specify multiple audio-processing plugins and their parameters. These presets include recommended processing for particular types of instruments (e.g. “Clarinet”) and abstract settings (e.g. “Ghost World”). Instrument-track “channel strip” presets specify synthesizer parameters in conjunction with additional audio-processing plugins and their parameters. Automated-drummer presets specify not only drum kits sounds and audio-processing parameters, but also the drumming style for an algorithmic drummer. Project presets are song templates that have specified audio tracks and instruments according to what one would expect for a particular genre of music (e.g. “hip-hop” or “songwriter”).

GarageBand also takes other approaches to simplifying audio production. After selecting a preset, GarageBand hides the low-level parameters of the audio-processing tools and synthesizers and instead shows a simplified interface with a reduced parameter set [**5, 7**]. GarageBand further lowers the barrier to audio production and composition by providing many prefabricated loops (audio recordings that can be seamlessly repeated), which are also named in similar fashion as the presets. These simplifying features make GarageBand a very easy tool to use. According to Madonna and Haim producer Ari Rechtshaid, GarageBand is able to make “anyone who buys an Apple computer a producer” [**184**].

GarageBand’s preset and drag-and-drop driven production may lower the barrier to audio production, but again, these simplifications come at a cost—they limit the creativity of users. A 2015 Pitchfork Article titled *Democracy of Sound: Is GarageBand Good for Music?* states that “according to several musicians, newer versions of GarageBand make it harder to innovate and customize, showing that there’s a fine line between a program that’s accessible and one that’s too accessible. **‘You feel like you’re being told what to do now’** (Prince Harvey)” [**184**]. Speedy Ortiz’s Sadie Dupuis said that **“I feel like the new GarageBand is devaluing my intelligence**

in a way" [**184**]. Even Grimes, whose breakthrough album was created using GarageBand, became frustrated with its limitations and said **"It really can't do anything... There's not a lot of stuff in GarageBand that's good"** [**184**]. Therefore, GarageBand doesn't meet one of Shneiderman et al's fourth criteria[1] for creativity support tools: "design with low thresholds, high ceilings, and wide walls" [**170, 147**]. GarageBand may provide easy entry to novices ("low threshold") because of its preset-dominated design, but its limited and inaccessible parameter space constrains its expressibility (i.e. lacks "high ceiling").

1.2. Problem Statement

1.2.1. Overview

> "I open programs like Ableton and sort of stare mouth agape at the screen" (Dee Dee of the Dum Dum Girls [**184**])

As audio technology has become more powerful and complex, it has also become more integral to the music-making process. The mastery of complex tools has become as important for composition as musical knowledge [**90**]. Unfortunately, it takes considerable amount of technical knowledge and experience to use these complex tools effectively, and these barriers continue to exist well into the digital age of audio production. Simpler interfaces exist, but they limit the creativity of the user and force the user to conform to the limited ideas of the software designers.

[1] Shneiderman et al's four criteria for creativity support tools are 1) *support exploratory search*, 2) *enable collaboration*, 3) *provide rich history keeping*, and 4) *design with low thresholds, high ceilings, and wide walls*

We need to rethink the interaction paradigms of audio production tools to provide usable perceived affordances [127, 56, 115] to users who don't have the experience, knowledge, or time to use powerful audio production tools effectively. Such users need to be able to communicate their ideas to audio production tools without having to use low-level technical parameters that they do not understand. In order to support creativity, these new interaction paradigms should also support interfaces that uphold the design principles of creative support tools: "low thresholds, high ceilings, and wide walls" [169].

1.2.2. Challenges

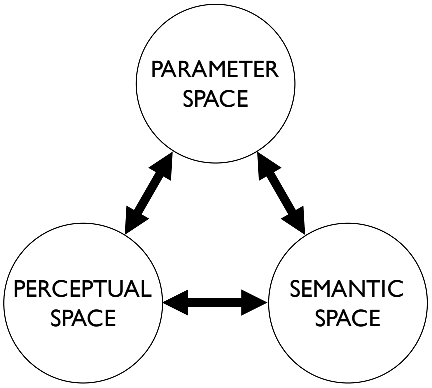


Figure 1.12. Mapping between the parameter space, perceptual space, and semantic space is one of the key challenges of rethinking the interaction paradigms of audio tools to provide usable perceived affordances to novice users.

Developing new audio production interfaces that have perceived affordances for novices and also support creativity is challenging. Reparameterizing these interfaces to provide perceived affordances for novices requires mappings between the *parameter space* of the audio production tools, the *perceptual space* of the user's auditory system, and the *semantic space* of the user's goals (see Figure 1.12). The parameter space of an audio production tool is the space defined by the low-level technical parameters that I described in 1.1. The number of parameters is often in the tens of parameters for audio processing tools but can easily be above one-hundred parameters for synthesizers such as Apple's ES2 synthesizer in Figure 1.1. The perceptual space of the user's auditory system is the space that describes how we perceive sound. This space includes dimensions of pitch, loudness, and timbre that change over time. Timbre is less understood than pitch and loudness [**123**]. It is defined as "That multidimensional attribute of auditory sensation which enables a listener to judge that two non-identical sounds, similarly presented and having the same loudness, pitch, spatial location, and duration, are dissimilar" [**2**]. In other words, it is all of the attributes of a sound that don't fall into our more well defined dimensions (loudness, pitch, spatial location, and duration) as timbre. Therefore, timbre is defined by what it is not rather than by what it is, and it is notoriously difficult to analyze [**97**]. Lastly, the semantic space of the user's goals is the space of higher-level meaning that users may use to describe their goal. For instance, a composer may describe their desired mix as "distant" and "velvety".

Solving this problem also requires semantic and perceptual data from human listeners, which can be difficult to obtain quickly. Because of the temporal quality of audio, it is more difficult to obtain labeled sound data than it is to obtain labeled image data. This difficulty results in a paucity of data for mapping these spaces. In addition, a listener's semantic interpretations of sound may be unique to that individual, and their perception of sound is dependent on their listening environment. If an audio production tool needs to learn a mapping that is unique to an individual, they must obtain training data and perform learning algorithms in a timely manner so that the user maintains their creative flow state.

Adding to the complexity, users' goals and data labels may also be noisy at times. Some audio production novices are also musical novices. These users may provide noisier training labels since their listening skills may not be as refined and their desired audio concepts may not be as clearly defined as experts. In addition, regardless of their experience, it is also common for a teacher's understanding of a concept to evolve and change as they teach. This is especially common for creative tasks—preferred goals and methods can and should shift during the creative process. This can be problematic for interactively training a machine learning system to assist a creative task such as audio production. Algorithms typically presume a constant goal and treat inconsistency in training data as unwanted noise. "Creative types" typically don't understand the internals of learning algorithms and cannot compensate for the weakness of the algorithms. We must develop methods that are better able to handle noisy training data that can possibly represent a shifting goal or concept as well. Ideally, these approaches should incorporate a training paradigm that even novice, non-technical users can use effectively.

A note regarding *novices*

In this work, I define *tool novices* [**33**] as users of audio production tools that lack the experience and knowledge of audio production tools, and I define *domain novices* [**33**] as users that lack the experience and knowledge of musical practice. Unless otherwise noted, when I use the unmodified term "novice" I will be referring to *tool novices.*

1.3. Approach

My approach to this problem is to look at the way that novices already communicate audio concepts (i.e. a particular sound, type of sound, or the "character" of a sound) to other people and then to build computational tools that understand and support those forms of communication. From my own experience and informal interviews with audio professionals, I have found that novices typically communicate audio concepts using:

(1) *Descriptive language*—e.g. "warm", "muddy", "tinny"
(2) *Examples* such as *vocal imitations* (e.g. "<*bwowowowowowow*>") and *existing audio recordings* (e.g. playing a recording of a particular sound or a recording of a song)
(3) *Evaluative feedback*—e.g. "more like this, less like that"

For example, in one interview a sound designer recalls a film director saying they wanted a "more *analog* <*ka-chunk*> *sound* of her flipping a switch". Here, <*ka-chunk*> was a vocal imitation of their desired sound and *analog* was a descriptive modifier that further defined the desired audio concept they wanted for the switch. She also recalled how clients often describe sounds with other descriptive terms such as "meaty", "thicker", "dull", "organic", etc. or referential descriptive terms such as a "John Bonham drum sound." Similarly, another audio professional recalled how people would communicate overall recording sound by bringing in examples of other recordings.

Other researchers have described similar observations of the interactions of studio engineers and musicians [**90, 91, 140**]. Porcello, an anthropologist, systematized his observations into five distinct communication methods:

(1) Lexical onomatopoesis—descriptive words that have partial acoustic resemblance to the sound they are describing (e.g. ‘hollow’, ‘ring’, ‘muffling’)

(2) ‘Pure’ metaphor—descriptive words that do *not* have acoustic resemblance to the sound they are describing (e.g. ‘tight’, ‘boxy’)

(3) Singing/vocable—i.e. vocal imitation

(4) Association—referring to other musicians, recordings, sounds, time periods, etc.

(5) Evaluation—“signifying agreement on sonic goals”

In contrast to these methods, the methods I identified are grouped by interaction modality, but Porcello’s methods are contained in these modalities and further support the communication methods I identified. *Lexical onomatopoesis* and *‘pure’ metaphor* are both subcategories of *descriptive language* of sound, and it can be difficult to distinguish between the two subcategories as well. *Singing/vocable* is a type of *example* that I refer to as *vocal imitation*—in my work, I refer to examples such as vocal imitation, which have some but not all of the characteristics of a desired sound, as *soft-examples*. *Association* maps into multiple of my communication methods. For example, using a particular sound or recording for communication maps on to my method of *examples*, and referring to a time period, e.g. an “80s snare sound”, would map on to *descriptive language*. Lastly, *evaluation* maps on to *evaluative feedback*.

My approach to rethinking the interaction paradigms of audio production tools is to build audio production tools to which users can communicate their desired audio concepts using similar methods as they communicate audio concepts to other people: *descriptive language*, *examples*, and *evaluative*

feedback. By using communication methods that users are familiar with, powerful audio production tools can provide usable perceived affordances to novice users who don't have the experience or knowledge of audio production tools. In terms of Schneiderman's design criteria [**171**] for creativity support tools, the goal of using these communication methods is to build tools with *low thresholds* and *high ceilings.* Building systems that can interpret these forms of communication requires mapping from these new input types to the low-level parameter space of audio-production tools. Learning these mappings from an individual can be time consuming for the user. Therefore, to hasten online communication with the user, I develop methods to learn mappings offline from the crowd when possible.

1.4. Dissertation Scope

My goal as a researcher is to unleash creativity by making complex audio tools accessible to everyone. However, there are many tasks in modern audio production: recording, audio editing, sound design, mixing, mastering. These tasks involve many types of tools. There are also many design principles that are also important to support the creativity of novices: support exploratory search, enable collaboration, provide rich history keeping, and design with low thresholds, high ceilings, and wide walls [**171**]. This dissertation does not address all audio production tasks or all tools, nor does it address all of the design criteria of creativity support tools. Rather, it focuses on 3 types of interaction (*descriptive language*, *examples*, *evaluative feedback*) to enable novices to communicate audio concepts to audio production tools with *low thresholds* and *high ceilings.* In addition, since examples of recordings and sounds are often not readily available, in this dissertation, I focus primarily on *vocal imitation* for the *example* method of audio concept communication to software. All of the proposed interaction methods also have their practical limitations. For each of these interaction types, I conduct research that advances this interaction type for audio production tools. This may be an actual audio production interface or an advance for mapping from that interaction type to low-level parameter spaces. Therefore, this dissertation does not provide a complete solution to our general problem of rethinking audio production tools to support creativity in novices. However, it is a significant step forward. The contributions of this dissertation can make some audio production tasks easier and enable some users who may have otherwise given up in frustration express their creative ideas.

1.5. Summary of Contributions

This dissertation explicitly looks to the methods with which novices communicate audio concepts to other people as a way to build affordances into audio production tools for novices. Based on their products, it seems that commercial audio production tool manufacturers only think in terms of low-level audio processing parameters and presets. To my knowledge, they have not explicitly thought about the methods that novices use to communicate audio concepts nor the mapping between parameter, perceptual, and semantic spaces (see Section 1.2.2). The contributions of my dissertation address these oversights.

In the remainder of this section, I summarize the outline of the dissertation and its main contributions.

(1) In Chapter 2, I present my work on communicating audio concepts to software with descriptive language. I propose a novel method for learning *actionable* “dictionaries” of audio descriptors for audio-production tools—a map from audio descriptors to audio concepts. This method allows a population of users to define perceptually relevant representations of audio-descriptors by simply listening to and evaluating audio examples. With these dictionaries, we can:

    (a) Use descriptive language for controlling audio production tools, building affordances into these tools for novices

    (b) Build two-dimensional descriptor maps for exploring audio production tools

    (c) Translate audio descriptors in the audio sense between languages

(2) In Chapter 3, I present my work on communicating audio concepts to software with evaluative feedback. I present the first audio-specific annotator model for aggregating audio evaluations from a crowdsourced population of listeners in varied conditions. With this methods and its accompanying software, we can:

    (a) Obtain and aggregate labels for learning audio concepts from a desired population (e.g., novices)

    (b) Evaluate audio algorithms on a population of listeners faster and more easily than using lab-based listening tests

    (c) Evaluate creative output (e.g. compositions, mixes, etc.) on a population of listeners

(3) In Chapter 4, I present research on communicating audio concepts to software with vocal imitation. I present a novel method for mapping vocal imitations to their referent audio. We can use this method to:

   (a) Program synthesizers using vocal imitation and evaluative feedback, building affordances into these tools for novices
   (b) Search sound effects databases with vocal imitation and evaluative feedback

(4) In Chapter 5, I present additional research on communicating audio concepts to software with vocal imitation. I present the first comprehensive and largest dataset of vocal imitations and labels. We can use this dataset to:
   (a) Investigate the strengths and weaknesses of communicating audio concepts with vocal imitations
   (b) Learn what types of audio concepts can be effectively communicated between people via vocal imitation
   (c) Learn the acoustic characteristics of audio concepts that can be effectively communicated between people
   (d) Clarify what software applications this method of communication can be used for
   (e) Learn more robust mappings from vocal imitations to audio concepts
   (f) Provide training data and a human performance baseline for query-by-vocal-imitation systems

1.6. Broader Impact

While the motivation of this dissertation is to make audio production more accessible to novices, the impact of this research is broader than that one problem. Audio/timbre-specific communication is a type of natural communication that has been neglected by speech recognition and natural language processing. Filling this gap can further machine intelligence and open up new interactions and audio applications.

Audio production tools are also not the only tools with interfaces in which users must express creative goals in the form of low-level parameters. Graphics, image, and video production tools share this characteristic as well. These complex tools also require technical and theoretical knowledge to use them effectively and could possibly support novice users by enabling communication of user goals via descriptive language, examples, and feedback. Therefore, while I focus on audio production in this dissertation, the methods presented in this dissertation could be adapted for other creative domains as well.

In addition to audio production for novices, this research can impact:

- *Audio production for visually impaired.* Audio production tools are very graphical and some are not accessible by screen readers. By enabling users to communicate audio concepts to software using language, vocal imitation, or evaluative feedback, users can bypass complicated graphical displays of low-level parameters.
- *Audio search.* The research in this dissertation on vocal imitation can also be used to search sound effect databases used by sound designers for games and film. For example, this research could help a sound designer find the "squeaky door" sample for a horror scene.
- *Music search (timbre-based query-by-humming (QBH)).* The research in this dissertation on vocal imitation could also be used to add timbre matching into QBH search engines—search engines which retrieve musical works based on humming

or singing a melodic line from the work. This could help users distinguish between songs with similar melodies but played on different sounding instruments.

- *Evaluation of audio algorithms or creative audio output (e.g. mixes).* The label aggregation method outlined in Chapter 3 can be used to perform subjective listening tests to evaluate audio algorithms or creative output (e.g. test out a few different mixes with a listening group)
- *Crowdsourced quality evaluation of non-audio media.* The Audio-Evaluator model presented in Chapter 3 primarily uses audio-specific predictors for reliability, but if reliability predictors for other domains can be constructed (e.g. estimating viewing conditions for image/video evaluation), this model form can be used for aggregating evaluations for those domains as well.
- *Hearing aid tuning.* The research in this dissertation on communicating audio concepts to software with descriptive language could aid novices in tuning hearing aids. Hearing aids are typically tuned by an audiologist in a clinic, but using the research in this dissertation, hearing aids could be tuned for different contexts and environments using descriptive language or a two-dimensional descriptor map.
- *Other media production tools for novices.* Other media production domains (e.g. graphics, images, video) have similar problems in which the tools are controlled by complex parameters with which novice users are unfamiliar. The descriptive language and evaluative feedback communication methods could also be used for these domains, and the mapping methods used in the dissertation could be adapted for those domains as well.

CHAPTER 2

Descriptive Language: SocialEQ

2.1. Overview

In this chapter, I present my work on communicating audio concepts to software with descriptive language (e.g., "make the vocals *warmer*"). I propose a method to build an audio-sense dictionary of actionable audio-equalization concepts. The "definitions" in this dictionary are indexed by descriptive terms for audio equalization, and these terms are defined not simply by other words but rather with a probabilistic frequency-domain representation that describes the descriptive term in the audio sense (see Figure 2.1). These audio-specific representations can be used to semantically control an equalizer, calculate similarity between descriptive terms for audio, translate the audio-sense of the descriptive terms between languages, and lastly to build two-dimensional semantic map-based equalization interface. The definitions in the dictionary are taught to the system by a population of novices using a simple evaluation task. Therefore, I use one of the communication methods mentioned in Section 1.3—*evaluation*—to teach a mapping to a semantic space so that novices can use another, quicker one of the communication methods—*descriptive language*. The methods outlined in this chapter enable the construction of audio equalizers and other audio production tools that respond to the descriptive language of novices and enable them to overcome the technical barriers to audio production.

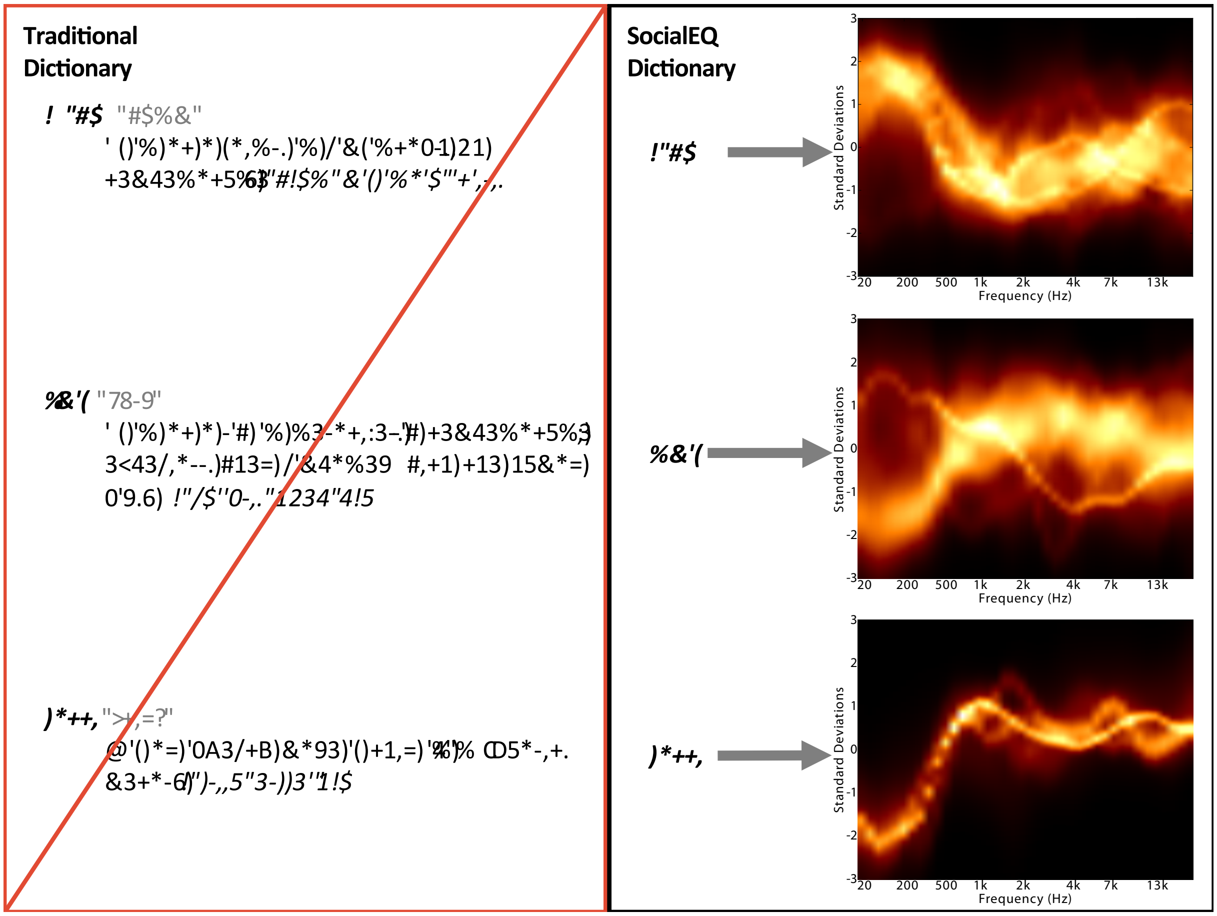


Figure 2.1. We are interested in building a dictionary of actionable audio concepts. Therefore, the definitions we learn are not words, but rather audio-sense definitions with perceptually grounded representations.

The work described in this chapter was presented at the International Society for Music Information Retrieval Conference [**18**] and the Collective Intelligence Conference [**21**].

2.2. Introduction

As mentioned in Section 1.3, one way in which people communicate audio concepts to each other is with *descriptive language*, and one of our goals is to build systems that understand this type of communication in order create affordances in audio production interfaces for novices. In order to understand this type of communication, we must first develop methods to learn the meanings of audio descriptors that a novice might use.

In this chapter, I present a method to learn the meanings (i.e. audio concepts) of descriptive terms for audio processing from a population of novices. I focus on the audio equalizer as an example audio production tool, but this method can be adapted to learn meanings of descriptive terms for other tools as well—both for audio and other types of media production. This method is also not limited to novices; it can be used to learn from any population which can perform the simple task of rating how much a descriptive term describes an audio stimulus. The audio concepts we learn for these descriptive terms can be used to map from the high-level semantic space to the low-level parameter space of an equalizer. They can also be used to calculate similarity between descriptors and therefore to build two-dimensional semantic maps and to translate between languages in the audio sense. I will discuss all of these uses in this chapter.

2.2.1. Audio Equalization

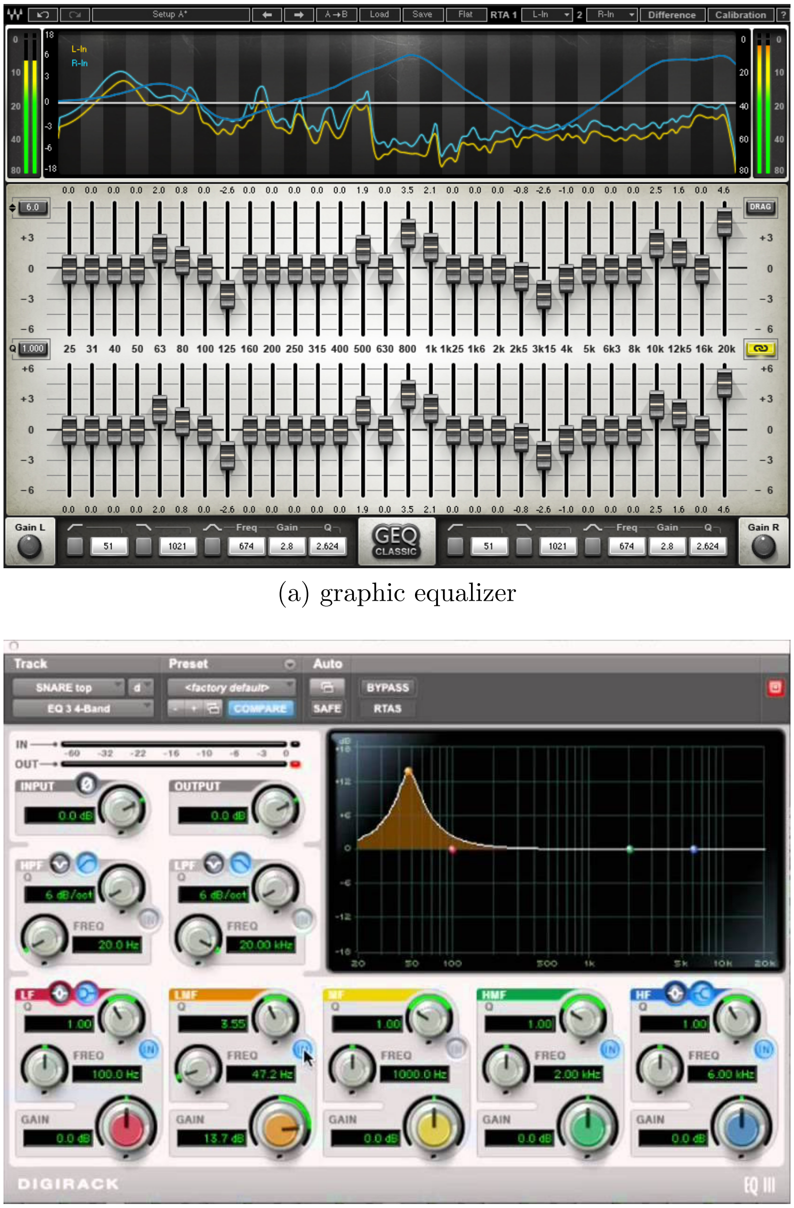
While the method can be used for other audio processing tools in addition to equalizers, equalizers themselves are very important to audio production tasks such as mixing, mastering, and sound design [**130, 83**]. An equalizer is an audio processor that alters the frequency balance of an audio signal using filters. The equalizer was first developed as a tool to correct for the fidelity loss in passive transmission of telephone signals; the equalizer "equalized" the output to the input–hence the name [**83**, 205]. However, in audio production it's used for much more than "equalizing" one signal to another.

According to Izhaki in his book *Mixing Audio* [**83**], there are four principal objectives for most mixes: *mood*, *balance*, *definition* and *interest*, and equalization plays a role in all of them [**83**, 58]. However, it's not only important, it's also hard. Izhaki states that "understanding frequencies and how to manipulate them is perhaps the greatest challenge mixing has to offer" [**83**, 205], and some say that "frequency treatment is half of the work there is in a mix" [**83**, 60].

Figure 2.2 shows two typical types of equalizers: a graphic equalizer and a parametric equalizer. Both people who are ignorant and knowledgeable about audio may recognize the graphic equalizer from consumer home audio equipment. In its consumer form, it allows listeners to graphically adjust the frequency range, over five to seven frequency bands. However, it was determined that typical consumers (i.e. novices to audio equipment) don't "significantly understand the meaning and application of equalization enough to benefit" [**188**], and therefore manufacturer's often reduce this control down to a single "loudness" button or a few genre-specific equalization curves (e.g., "rock", "classical", "pop") [**188**]. However, these buttons do not map onto the typical goals of even consumers and are often misused [**188**]. In audio production software, equalizers are usually even more complex—either 30–40 frequency band graphical equalizers and complex parametric equalizers as shown in Figure 2.2.

2.2.2. Existing Approaches to Make Audio Equalization Easier

Given the importance of equalization and the difficulty of understanding how to achieve a particular mixing objective with equalizers, researchers have sought to develop easier methods to effectively control equalizers. Bitzer et al [**11**] developed an algorithm to



(a) graphic equalizer

(b) parametric equalizer

Figure 2.2. Screenshots of a typical graphic equalizer (Wave's GEQ Graphic Equalizer) and a typical parametric equalizer (the built-in equalizer in Avid's Pro Tools). Sources: `http://www.waves.com/1lib/images/products/plugins/full/geq-graphic-equalizer-classic.jpg` and `https://i.ytimg.com/vi/Emq_O19q_m8/maxresdefault.jpg` (downloaded July 2016)

detect the salient frequencies of a particular audio signal. This approach could be used to reduce the parameter space of an equalizer, allowing the user to focus on the manipulation of a small set of frequencies. However, as consumer audio manufacturer's learned, non-technical users may find the manipulation of even a few frequencies confusing [**188**]. Many researchers have sought to solve this problem by using "automatic" algorithmic mixing techniques [**138, 6, 108, 62**] that typically use optimization- or heuristic-based approaches to tune the parameters of an equalizer and other mixing tools. In regards to Izhaki's four mixing objectives [**83**], the objectives of algorithmic mixing approaches are typically to *balance* the loudness between tracks and to reduce frequency overlap (e.g., auditory masking) in order to increase the *definition* of the individual tracks. However, the automatic mixing approaches eliminate the user's control and therefore their creative expression. A user may actually want to *decrease* the definition of certain tracks, blending two tracks together. A user may also want to use an equalizer to achieve Izhaki's other two mixing objectives: *mood* and *interest*. However, automatic approaches typically don't allow for such user expression.

In contrast to the automatic approach, both research and commercial audio software have attempted to give control to the user by supporting some form of descriptive language input, but the choice of vocabulary and the correct mapping is paramount to the success of this approach. For example, most commercial audio software tries to incorporate descriptive language by simply using named preset settings. However, not only are presets often poorly named, but they often use the vocabulary of *experts*. In a slightly different approach, Reed [**145**] built an equalization interface that was controlled by descriptive terms but was also signal-dependent (e.g., "brighter" was applied differently depending on the input signal). However, his system uses a nearest neighbor approach that requires a lot of data to truly be signal-dependent, and it is also trained by *experts* and hence uses the language of experts. Similar to our goal, Mecklenburg and Loviscach built an actionable two-dimensional map of equalization descriptors [**117**], but again, their system was trained by *experts*. Experts trained these systems by using traditional equalizer interfaces (e.g., Figure 2.2) and associating different parameter settings with audio descriptors.

Unfortunately, novices and experts differ in their vocabulary used to describe sound, as noted by Porcello in his observational studies in recording studios [**140**]. In fact, Mecklenburg and Loviscach also recognized this when evaluating their system. They found that experts liked using their two-dimensional equalizer interface but amateur musicians did not—they didn't agree with the vocabulary of the system.

While not a system for equalization, Ethington and Punch's SeaWave [**45**] was an early (1994) audio production system that utilized descriptive language that *could* be taught by novices. The SeaWave applied transformations to existing additive synthesizer sounds using descriptive terms (e.g., more "resonant"). The system was trained by varying high-level parameters of the sound (e.g., harmonic density) and having listeners rank how much a particular descriptive term from a predefined vocabulary described the variation. It would then correlate the terms to the parameters and use this information to transform the sound given one of the descriptive terms in the system. A later synthesis system was developed by Johnson and Gounaropoulos that tried to automate aspects of the SeaWave's process using machine learning [**88, 60**]. Unfortunately, it is unclear from these papers how effective either of these systems are, and both of these systems make the assumption that everyone agrees on the meanings of the descriptive terms—an assumption which is invalid.

2.2.3. Understanding Audio Descriptors and Their Acoustic Correlates

I believe that audio production tools to which users communicate their audio production objectives with descriptive language (e.g., "make the vocals *warmer*") can help novices overcome the barriers of audio production tools such as equalizers. When using language to control a system, its essential to train the system with the vocabulary of the users of the system. For our goals, the tools need to understand the agreed-upon vocabulary that novices use, not experts. In addition, these audio production tools must be able to tell whether the stated goal is achievable for the selected tool (e.g., making the violin "warmer" with a panning tool does not make sense). It must also know what actions need to be taken, given the correct tool ("Use the parametric equalizer to boost the 2-4 kHz and the 200-500 Hz bands by 4 dB"). Further, the tool should be aware of possible variations in the mapping between words and audio among users (Bob's "warm" $\neq$ Sarah's "warm"), and the tool should be aware of which words are synonymous.

Such tools are difficult to implement because there currently is no universal dictionary of audio terminology that is defined both in terms of subjective experiential qualities and measurable properties of a sound. Tools built from text co-occurrence, lexical similarity and dictionary definitions (e.g., WordNet [**118**]) are fundamentally different in their underpinnings and do not address the issue of how words map to measurable sound features.

There are some terms relating to pitch ("high", "low", "up", "down") and loudness ("soft", "loud") that have relatively well-understood [**168, 70**] mappings onto measurable sound characteristics. Some terms of art used by recording engineers [**78**] describe effects produced by recording and production equipment, and are relatively easy to map onto measurable properties. These include "compressed" (i.e. a compressor has been applied to reduce the dynamic range of the audio) and "clipped" (i.e. the audio is distorted in a way characteristic of an overloaded digital recorder). These terms are not, however, widely understood by either musicians or the general public [**188**]. The vocabulary used by experts also takes time to learn. In fact, in some recording engineering schools, there are entire courses devoted to teaching students how to communicate the perception of timbre using agreed-upon language [**105**].

Numerous studies have been performed over the last fifty years in the hopes of finding universal descriptive terms for sound that map onto a set of canonical perceptual dimensions [**176, 61, 113, 202**]. Also, in the last decade or so, many researchers coming from backgrounds such as recording engineering [**78**], music composition [**173**] and computer science [**160**] have studied the space of terminology, seeking a universal set of English terms for timbre. Researchers in acoustic signal processing and perception continue to seek English taxonomies for descriptive terms for timbre. Some seek general taxonomies of descriptive sound descriptors [**160**]. Others are focused on adjectives for the timbre of particular instruments [**107, 38**].

Such studies typically start by determining a vocabulary of natural descriptive terms by performing a survey. This vocabulary is then used in a second study, where participants evaluate sounds in relation to the terms. These are then mapped onto machine-measurable parameters, such as spectral tilt, sound pressure level, or harmonicity. Commonalities in word mappings are found between participants in the studies and then some small set of descriptive terms are proposed as relatively universal.

Despite the efforts of all these groups, a universal map of descriptive terms for audio remains an unachieved goal. I believe this is because the terms used range from widely-agreed-upon (e.g., "loud") to ones that have agreement within groups but not between groups (e.g., "warm") to idiosyncratic terms meaningful to sole individuals (e.g., Martha's "splanky").

In this work, rather than seek a set of universal underlying dimensions of timbre or the universal meaning of a descriptive terms in all auditory contexts, I seek an understanding of descriptive terms in a specific context: audio equalization.

As mentioned earlier, in audio engineering, there has been some work directly mapping equalizer parameters to commonly used descriptive terms [**117, 145**]. A problem with these approaches is that the mappings are trained by experts because they require knowledge of audio equalizers to perform the training. These approaches also make the assumption that all users would agree on the meaning of the trained audio descriptors—i.e., there is no variability that exists across users.

In contrast, Sabin et al developed a system to which a user can teach a personal and actionable audio-equalization concept by simply evaluating a series of 25 audio examples [**158, 156**]—i.e., using evaluative interactions, it learns a mapping from an audio descriptor to a personalized actionable controller (e.g., Bob's "warm" knob). Since the teaching procedure doesn't require knowledge of audio equalizers, even a novice can train the system. This method was later extended to learn personalized audio reverberation controls [**144, 156**]. In addition, further research showed that the amount of time required to learn the personal audio concepts could be reduced by over half [134, 133]. While user studies established the effectiveness of Sabin's approach, the method always requires training for a new descriptive term and the user must clearly know what they want, making it difficult to explore the audio equalization concept space. To address this, Sabin et al used the data from their user study of the system—personal audio-equalization concepts for 5 audio files and 4 descriptors ("bright", "tinny", "warm", "dark") from a population of 19 subjects—to build 2DEQ, a two-dimensional equalization map interface using the first two principal components of their data [157]. However, since this map was trained on just four descriptors, it captures only a fraction of the audio-equalization concept space.

2.2.4. Overview of Our Approach

In this chapter, I describe SocialEQ, a project to crowdsource a vocabulary of audio descriptors, grounded in perceptual data that can be mapped onto concrete actions by an equalization (EQ) to effect meaningful change to audio files. We use Sabin's learning algorithm to **learn hundreds of personal audio-equalization concepts from a population of hundreds of *novices***. We then aggregate the personal audio-equalization concepts into audio-equalization descriptor definitions. From these definitions, we can **determine which words have shared meanings for a population of novices**, allowing us to **map from the descriptive language of general novices to the controls on an equalizer**. We essentially use a simple but time-consuming communication method (evaluation) to learn a mapping for a simple and quick communication method (descriptive language). By utilizing an evaluation-based teaching procedure, we are not limited to the language of experts, unlike the methods by Reed [145] and Mecklenburg et al [117].

And by learning the meaning of audio descriptors and which terms are actionable by an equalizer and have agreement within a population, we can create audio production tools for supporting novices using fast communication with descriptive terms rather than slow, evaluative-based communication like the work of Sabin et al [**156**]. SocialEQ has been taught 388 distinct English words and 384 distinct Spanish words in a total of 2322 training sessions. By using this data of perceptually-grounded representations of descriptive terms, we can calculate similarity between the audio concepts described by the terms, translate the audio sense of terms between two languages (e.g., what is the audio sense of “warm” in Spanish?), and ultimately build two-dimensional descriptor maps that can be used for controlling equalizers and exploring the audio-equalization concept space.

2.3. The SocialEQ Task

SocialEQ is a web-based application that learns an audio-equalization concept associated with a user-provided audio descriptor. To deploy the software to a large audience for our data collection, we have implemented it as a web application using Adobe Flex and Drexel University’s Audio Processing Library for Flash (ALF) [**164**].

After agreeing to participate, we ask participants to “enter a descriptive term in the language in which you are most comfortable describing sound (e.g., ‘warm’ for English, ‘claro’ in Spanish, or ‘grave’ in Italian), pick a sound file which we will modify to achieve the descriptive term, then click on ‘Begin’.”

Participants are then given the choice of three different source audio files to manipulate: “Electric Guitar”, “Piano”, and “Drums”. All files were sampled at 44.1 kHz and 16 bits. The files all had constant sound (i.e., no breaks in the audio) and were presented

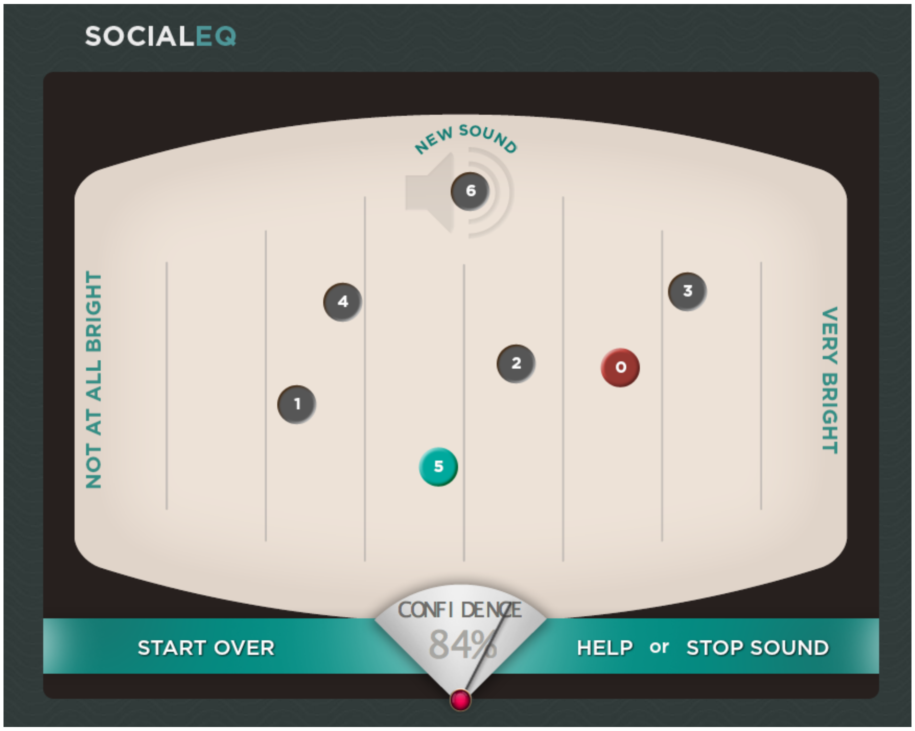
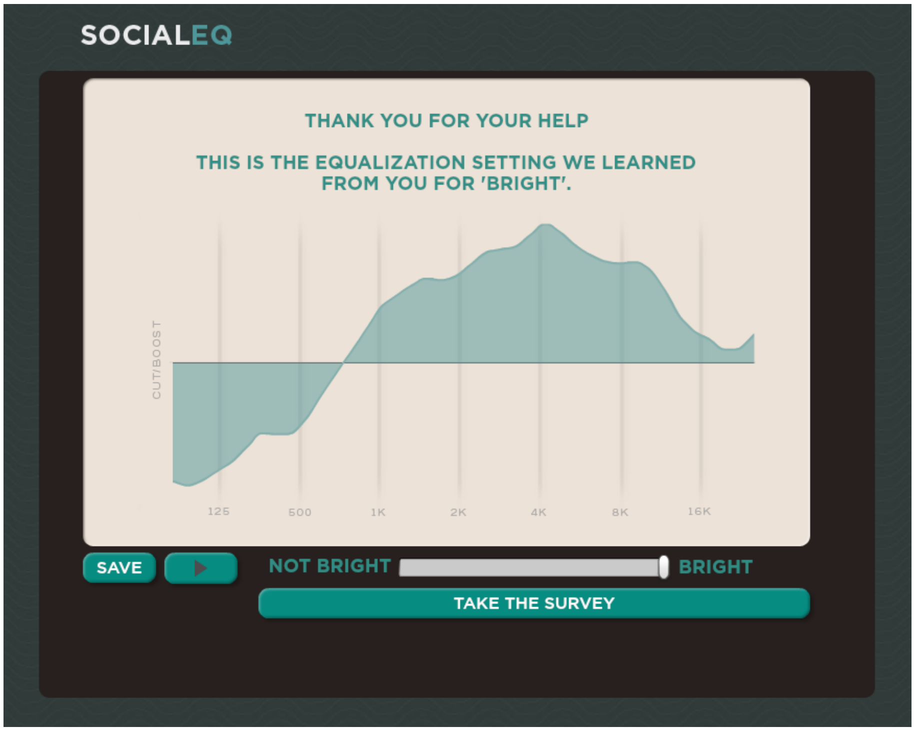


Figure 2.3. SocialEQ ratings screen. The user rates how well each audio example corresponds to the descriptive term they selected at the prior stage. Users rate an example by dragging the associated circle to the desired horizontal position. Vertical position does not matter.

in loops that were 8, 10, and 14 seconds long, respectively. All three sound files also had wide frequency bandwidths (see Figure 2.5).

Once a participant selects a descriptive term and a sound file, they are asked to "rate how 'your word' that sound is" using the interface shown in 2.3. The participant then rates 40 examples of the chosen audio file, each modified with an equalization curve. Fifteen curves are repeats of prior examples, to test for rating consistency. Figure 2.3 shows the interface for rating examples.


THANK YOU FOR YOUR HELP

THIS IS THE EQUALIZATION SETTING WE LEARNED FROM YOU FOR 'BRIGHT'.

Figure 2.4. The equalization curve and control slider learned for "bright" in a single session.

From these rated examples, the system learns the relative spectral curve—the relative boost/cut to apply each of 40 frequency bands—that represents the personal audio-equalization concept for the participant. These bands have equivalent recatangular bandwidth (ERB) [**58**] derived center frequencies. We use the method from [**156**] to learn the EQ curve from user responses (see Section 2.4). This method treats each frequency band as independent and correlates changes in user ratings with changes in gain for that band. Positive correlation indicates a boost, and negative correlation a cut. The result is a 40-band EQ curve for the descriptor learned from the participant. The system uses the learned equalization curve to build a personalized slider that lets the participant manipulate the sound in terms of their interpretation of the descriptive term (Figure 2.4).

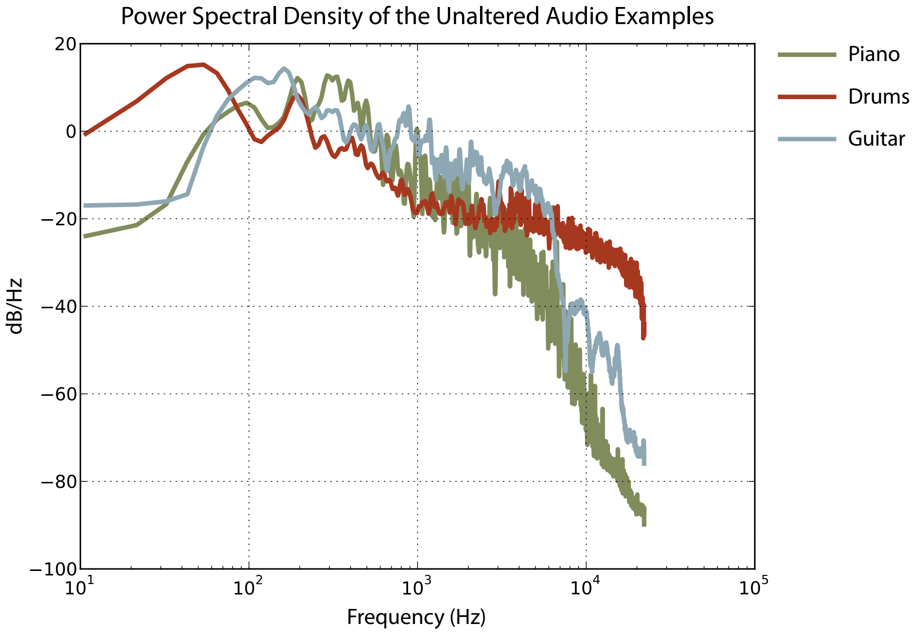


Figure 2.5. The power spectral densities of the audio examples before being filtered with the audio-equalization probe curves to create the stimuli.

After teaching SocialEQ a descriptive term and trying the personalized slider, participants were asked to complete a question survey that assessed their background, listening environment, and experience using SocialEQ.

2.4. Learning a Personal Audio Concept from Ratings

To learn the relative spectral curve representing a personal audio-equalization concept, we utilize the method introduced by Sabin et al in [**158**]. As mentioned in Section 2.3, users are presented a series of audio examples, each of which is processed with a different audio-equalization curve from a set of probe curves. The users are asked to rate how much their specified descriptive term describes each audio example.

To generate the audio-equalization probe curve set, 1000 probe curves were constructed by generating and concatenating 2-8 Gaussian curves with random amplitudes (-20 – 20

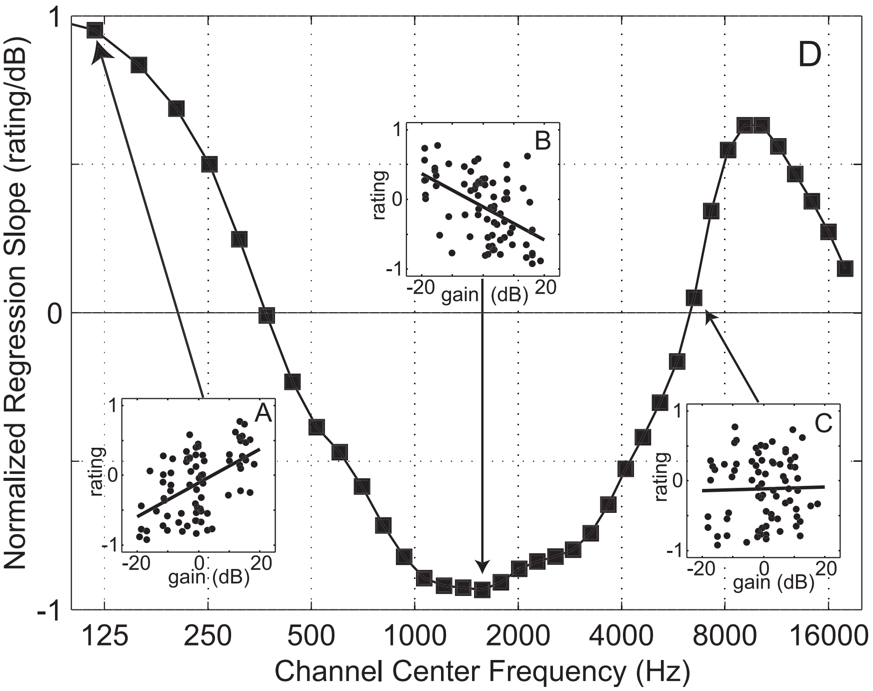


Figure 2.6. An example of a relative spectral curve for file/descriptor combination of drums/“warm”. Each point in the curve corresponds to a frequency-band weighting that is slope coefficient learned from a linear regression between a single user’s ratings and the frequency-dependent gain of the training example. Image from [**156**].

dB), bandwidths (5–20 frequency bands), and center frequencies. This larger set of 1000 probe curves was reduced to a diverse of 25 probe curves by greedily optimizing diversity while also ensuring that the distribution of within-channel gains is be comparable across channels.

Independently for each frequency band, a simple linear regression is performed between the user’s ratings of the audio examples and the frequency band’s gains specified in the audio-equalization probes used to generate the audio example (see Figure 2.6). The regression slope coefficients are then concatenated together to from the relative spectral

Table 2.1. Definitions of terms

descriptive term	A word describing audio (e.g., *warm*) taught to the system in at least one training session. We may also refer to this as a *descriptor* for brevity.
session	The complete process in which a participant teaches SocialEQ a descriptive term, tries the learned personal controller, and completes the survey.
relative spectral curve (RSC)	A set of relative gains (i.e., boosts or cuts) on the 40 ERB frequency bands. It is used to represent a personal audio-equalization concept. To make an RSC comparable, every RSC is normalized by putting gain values in standard deviations from the mean value of the 40 bands.
personal audio concept	An audio concept learned in a single session consisting one person's interpretation of a descriptive term (e.g., the session where Bob teaches 'warm' to SocialEQ). It is represented by an relative spectral curve (RSC).
descriptor definition	A general audio concept comprised of a set of personal audio concepts that all represent the same descriptive term. A definition may be vague or precise depending on how much agreement there is between personal audio concepts that share the descriptive term. Figure 2.9 shows *deep* and *sharp*.

curve. In SocialEQ, each curves is then standardized with itself—centered around the mean and divided by the standard deviation.

2.5. Building an Equalization Descriptor Map

In this section, we address the following questions in order to build an equalization descriptor map:

(1) What audio descriptors are actionable by an audio equalizer?

(2) How widely-agreed-upon is the meaning of an equalization descriptor?

(3) What equalization descriptors are true audio synonyms within a language?

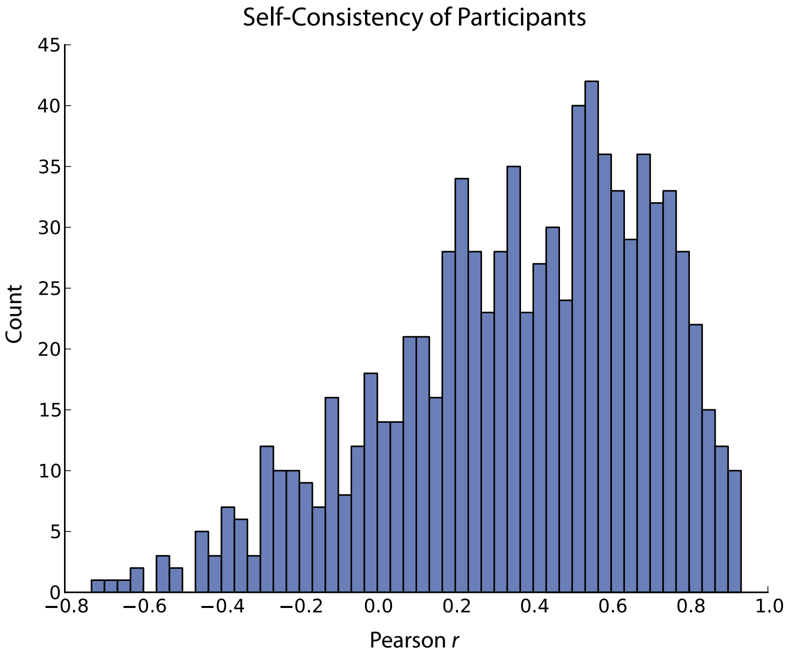


Figure 2.7. Histogram of participant's ratings self-consistency which was calculated by taking the Pearson correlation between the two sets of ratings of the 15 stimuli that were repeated.

2.5.1. Data Collection

For a data collection of this size, an on-site data collection was not feasible. We instead recruited participants through Amazon's Mechanical Turk (AMT). We had 633 participants who participated in a total of 1102 training sessions (one session per learned word). We paid participants \$1.00 (USD) per session, with the possibility of up to a \$0.50 bonus, determined by the consistency of their equalized audio example ratings. While we could not control for the quality of loudspeakers, over 92% of the participants reported listening over either headphones or large speakers (rather than small/laptop speakers).

2.5.1.1. Inclusion Criteria for Sessions. Before analyzing the results, we first removed sessions by participants who didn't seem to put effort into the task. The mean time to teach the system a single personal audio concept for a descriptor was 292 seconds ($SD = 237$). We removed all sessions where the participant completed the task in less than 60 seconds. We also removed all sessions where the participant gave the default rating for more than 5 out of the 40 examples. We also removed any session where the participant responded “no” to the survey question: “Was the listening environment quiet?”.

Recall that 15 of the 40 examples were repeats in any session. Using the ratings for the repeated examples, we can assess a participant’s consistency when teaching SocialEQ. We measured consistency using Pearson correlation between the ratings of the test and repeated examples. The median consistency across sessions was 0.41 (95% confidence interval (CI) [0.39, 0.44]) (see Figure 2.7).

Only sessions with consistency greater than zero were retained. This left 481 participants who taught the system in 731 sessions. Individual participants were allowed to teach more than one descriptive term to the system. The maximum number of descriptive terms a participant taught was 9.

2.5.2. Descriptive Term Analysis

In this section we provide an analysis of some of the data, but we have also made the data available for use by the research community at `http://socialeq.org/data`.

2.5.2.1. The descriptors. In the 731 sessions, there were 324 unique descriptive terms. The descriptors taught most frequently are listed in Table 2.2. Of the 324 words, 91 occurred two or more times. The most popular descriptor was *warm*, but note that there was a bias for participants to teach the system *warm* due to its use as the example in

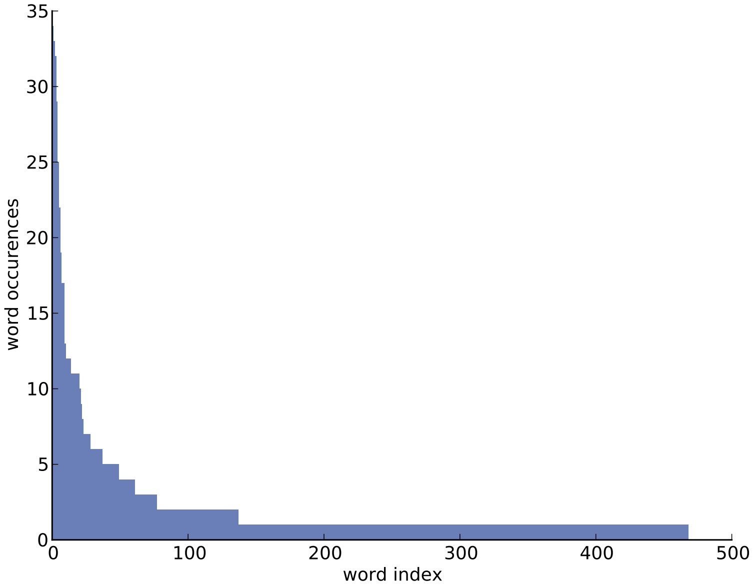


Figure 2.8. Word occurrence distribution of the SocialEQ descriptors.

Table 2.2. The 10 most frequently contributed descriptive terms

Rank	Descriptor	Sessions
1	warm	57
2	cold	25
3	soft	24
4	loud	22
5	happy	19
6	bright	16
7	harsh	15
8	soothing	14
9	heavy	11
10	cool	11

the instructions (see Section 2.3). The word occurrence distribution (see Figure 2.8) has a short head / long tail that is reminiscent of Zipf's law [**109**].

2.5.2.2. Representing audio-equalization concepts. In this paper, we make the assumption that participants judged each equalization example relative to the unprocessed

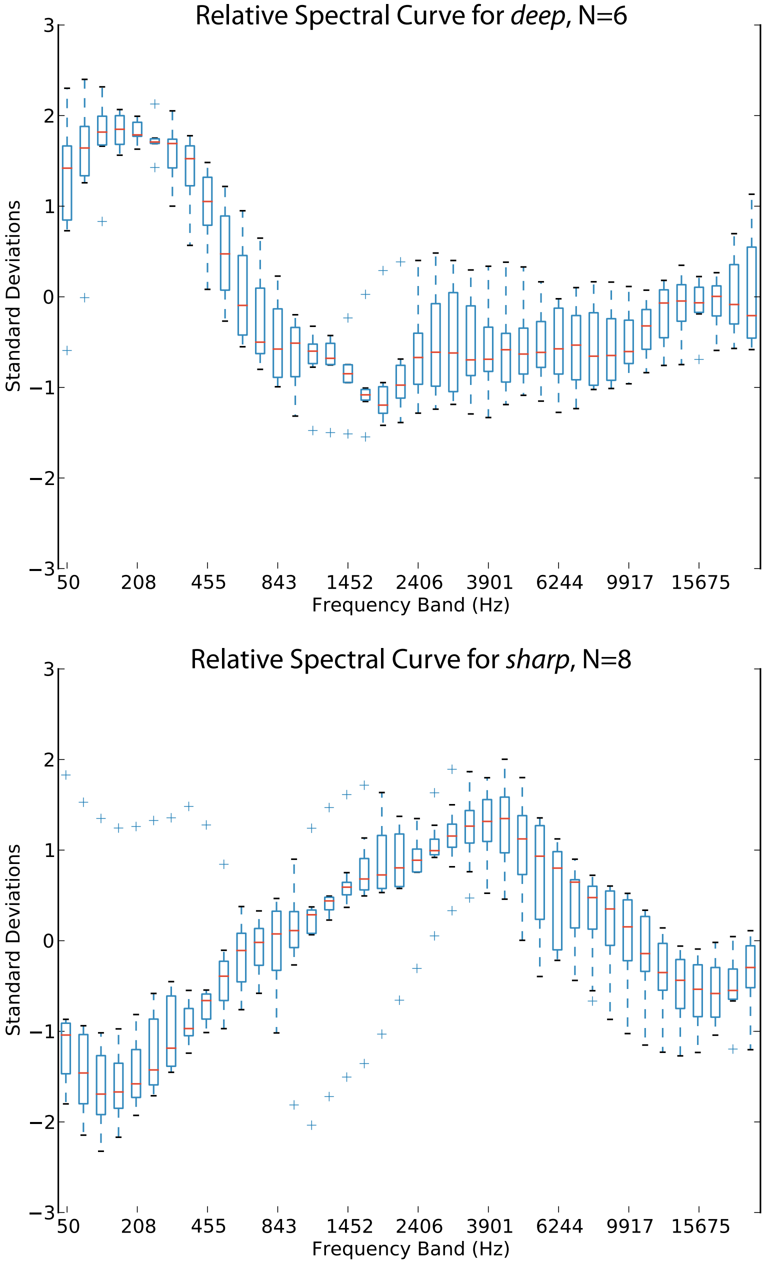


Figure 2.9. Per frequency band boxplots of RSCs for the descriptors *deep* and *sharp*, learned in 6 and 8 sessions respectively. In each ERB-spaced frequency band, the center line is the median, the box represents 50% of the data, the whiskers represent the remaining 50%. The pluses represent outliers.

Table 2.3. Top 10 equalization descriptors taught to the system by at least 4 people, ranked by *mean slider rating*, which ranges from -3 (Strongly Disagree) to 3 (Strongly Agree)

Rank	Word	Mean Response
1	relaxing	2.75
2	quiet	2.60
3	hot	2.50
4	hard	2.50
5	heavy	2.36
6	smooth	2.33
7	deep	2.33
8	bright	2.31
9	soothing	2.31
10	mellow	2.29

source rather than judging the absolute spectrum of each equalization example. We therefore have chosen to represent equalization concepts in terms of relative changes in each frequency band rather than the resulting spectrum after equalization. This allows us to compare equalization concepts from varying source material played on varying loudspeakers. In Figure 2.9, we show the distribution of RSCs collected for two example descriptors: *deep* and *sharp.* Each column shows the distribution of learned values for the corresponding ERB-spaced frequency band.

Note that except for one outlier participant, there was fairly high agreement for the meaning of *sharp.* This is interesting, since most musicians are taught that *sharp* relates to relative pitch, rather than the spectral characteristics of a sound. Our data indicate *sharp* also has other connotations that relate to timbre.

2.5.2.3. Actionable equalization descriptors. As stated in Question 1 in Section 2.2, one goal of this paper is to determine what audio descriptors describe goals achievable by an audio equalizer (i.e., the descriptive term is an equalization descriptor). One way to

Table 2.4. Top 10 descriptive terms ranked by the *agreement score* described in Section 2.5.2.4

Rank	Word	Agreement Score
1	tinny	0.294
2	pleasing	0.222
3	low	0.219
4	dry	0.210
5	metallic	0.195
6	quiet	0.188
7	deep	0.164
8	hollow	0.160
9	light	0.131
10	warm	0.130

answer this is to simply look at the *mean slider rating*: the mean response to the survey statement, "The final (control) slider captured my target audio concept." Participants were asked to respond on a 7-level Likert scale coded from -3 (Strongly Disagree) to 3 (Strongly Agree). Table 2.3 shows the 10 descriptive terms with the highest mean response to this question that were contributed by at least 4 participants.

The learning approach used by SocialEQ [**156**] has an inherent bias towards learning smooth equalization curves and has difficulty learning curves with narrow boosts or cuts or frequency relationships that are non-linear or dependent. Therefore, *mean slider rating* is a sufficient but not necessary condition to determine whether the descriptor is actionable by an equalizer.

2.5.2.4. Agreed upon equalization descriptors. The second question we would like to answer is "How widely-agreed-upon is the meaning of an equalization descriptor?". The meaning of an equalization descriptor, as embodied in the personal audio concept learned in a particular session, may vary significantly from person to person. We want to find which of these vary the least from person to person, or rather which descriptive terms have the most widely-agreed-upon meanings. To answer the question we looked at the total variance (i.e., the trace of the covariance matrix) of RSCs within a descriptor definition. This can be written simply as the sum of the variance in each RSC frequency-band for the personal audio concepts in a descriptor definition:

$$\text{trace}(\Sigma)_{descriptor} = \frac{1}{N}\sum_{k=0}^{39}\sum_{n=0}^{N-1}(x_{n,k} - \mu_k)^2 \tag{2.1}$$

where N is the number of personal audio concepts in the descriptor definition, k is the index of the frequency band, x is the RSC for personal audio concept n, and μ_k is the mean of frequency band k over the N personal audio concepts in the descriptor definition.

If we then divide the natural logarithm of the number of personal audio concepts (log(N)) by this value, i.e.:

$$agreementscore = \frac{log(N)}{\text{trace}(\Sigma)_{descriptor}} \tag{2.2}$$

we have an *agreement score* that takes into account both total variance and the popularity of the descriptive term. We used log(N) to linearize the number of personal audio concepts since the frequency with which a descriptor was taught to the system was distributed with a short head and long tail (see Figure 2.8). When we rank the descriptive terms by this score, we discover which terms have more agreement amongst the participants. The top ten descriptive terms ranked by this score are shown in Table 2.4.

2.5.2.5. Audio descriptor synonyms. To answer our last question, “What equalization descriptors are true audio synonyms within a language?”, we compared the descriptor definitions using a distance function.

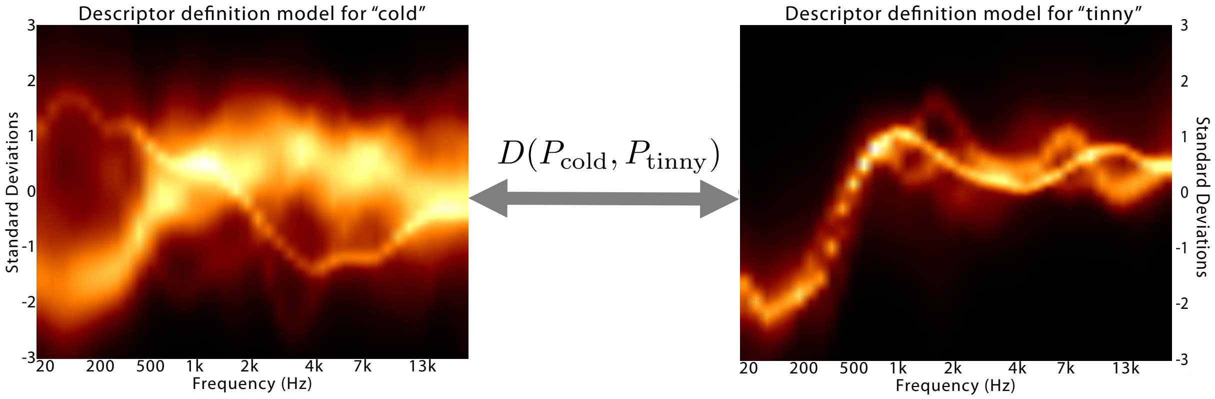


Figure 2.10. We calculate the distance between two descriptor definition models using symmetric KL divergence.

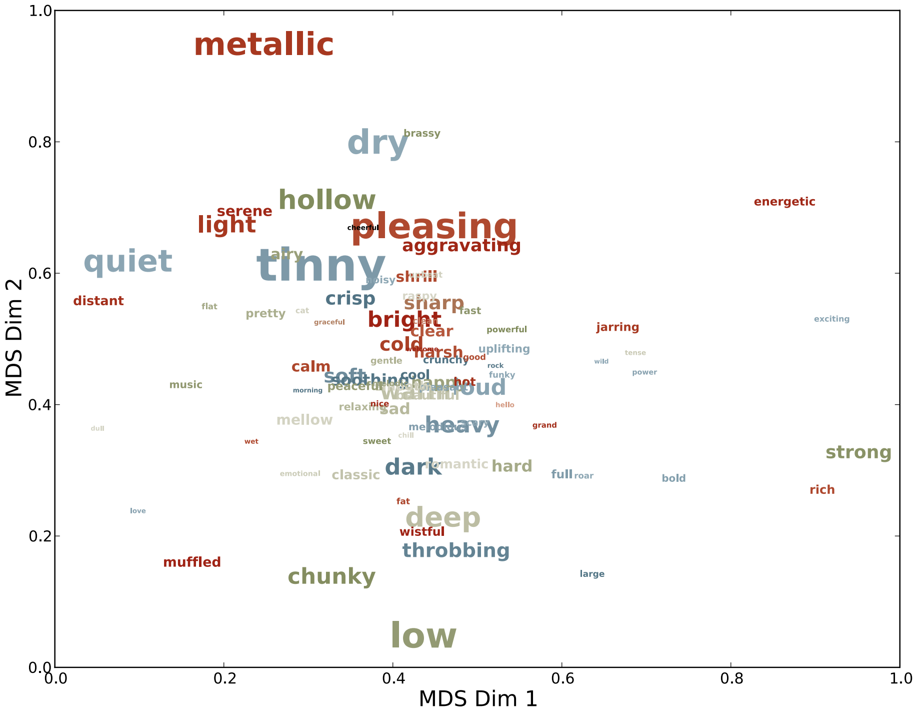


Figure 2.11. Two-dimensional equalization descriptor map build using multi-dimensional scaling (MDS) on the distances (see Equations 2.3 and 2.5) between descriptive terms. Their font size positively correlates to their agreement score from Equation 2.2.

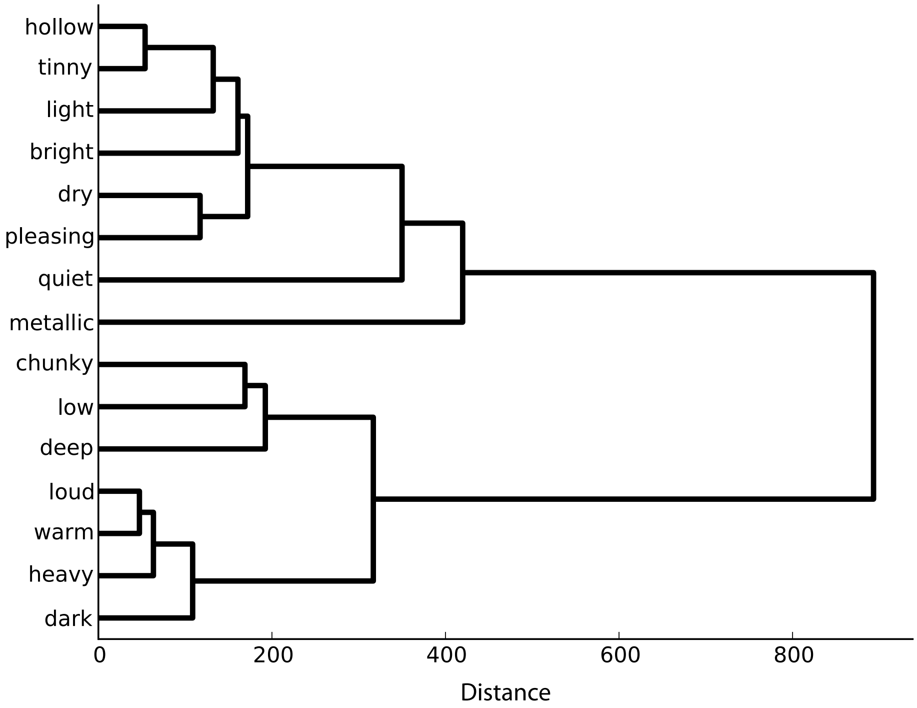


Figure 2.12. Hierarchical clustering of the 15 descriptive terms with the highest agreement scores.

Table 2.5. Synonyms of high agreement descriptive terms (see Table 2.4) found through comparing descriptor definitions

descriptor	synonyms
light	tinny, crisp
tinny	hollow, crisp, light, shrill, bright, cold, raspy
deep	throbbing, dark
hollow	tinny, dry, shrill, pleasing

To compare learned definitions, we wanted a distance measure that would: 1) allow for varying number of personal audio concepts per definition; 2) allow for multi-modal distributions; and 3) take into account the uncertainty of the definition. Therefore, we modeled each descriptor definition as a probability distribution over the personal audio concepts for the descriptor, and we then compared definitions using an approximation of the symmetric Kullback-Leibler divergence.

The steps to calculate the distance between two descriptor definitions are:

(1) Model each personal audio concept as a Gaussian distribution, $\mathcal{N}(\mu_n, \Sigma_n)$, where μ_n is the RSC of the personal audio concept and Σ is a diagonal covariance matrix in which the variance for each frequency-band is set by $\sigma_{n,k}^2 = (\sigma_k - \sigma_k r_n)^2$ where σ_k is the sample standard deviation of frequency-band k for RSCs of *all* descriptors, and r_n is the ratings consistency for the session that learned personal audio concept n. Here we are using the ratings consistency as a measure of the uncertainty of the personal audio concept, mapping a consistency range of $[0, 1]$ to a per-frequency-band variance range of $[\sigma_k, 0]$.

(2) Then model each descriptor definition as follows:

$$P(x) = \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{N}(\mu_n, \Sigma_n) \tag{2.3}$$

where $\mathcal{N}(\mu_n, \Sigma_n)$ is the distribution for the n^{th} user-concept.

(3) We then use a Mone Carlo method to calculate an approximate symmetric Kullback-Leibler divergence [**72**], $D(P_1, P_2)$, to compare two definition models, P_1 and P_2:

$$D(P_1, P_2) = D_{\mathrm{KL}}(P_1 \parallel P_2) + D_{\mathrm{KL}}(P_2 \parallel P_1) \tag{2.4}$$

$$D_{\mathrm{KL}}(P_i \| P_j) = \int_{-\infty}^{\infty} p_i(x) \, \log \frac{p_i(x)}{p_j(x)} \, \mathrm{d}x \tag{2.5}$$

Using this distance measure, we computed the distance between every pair of definitions. With these distances, we can map and visualize the relationships of definitions. In Figure 2.11, we placed the descriptor definitions in a two-dimensional space by using metric MDS [**13**], each definition is scaled by its agreement score so that well-defined descriptors are larger. To get a better sense of the relationships of the 15 high agreement descriptive terms listed in Table 2.4, we performed agglomerative hierarchical clustering using the "group average" algorithm [**63**] and plotted the dendrogram in Figure 2.12. From this you can see two clusters formed with descriptors one might typically associates as opposites: *bright/dark*, *quiet/loud*, *light/heavy*. From this we can also see relationships of descriptors which may not have been obvious such as *pleasing* being closely associated with *dry*.

In Table 2.5, we list a few high agreement descriptive terms along with their synonyms through comparing definitions. We considered two words synonyms if their distance was within the first percentile of all the pairwise distances. Also, only definitions that consisted of at least two personal audio concepts and had at least a 1.0 *mean slider rating* (as described in Section 2.5.2.3) were included.

2.6. Translating Audio Descriptors

Correctly translating adjectives that describe sound (e.g., "heavy", "tinny", "dry", "soothing") can be a difficult task [**203**]. Resources such as the Oxford English Dictionary (OED) [**141**], typically list the "audio sense" for only a small subset of the words commonly used to describe sound. For example, "warm" is a very commonly used sound adjective and the OED does not mention the audio sense. Directly translating the predominant (i.e., first) sense of a sound adjective into another language often results in an incorrect translation. For example, when "a warm sound" is typed into Google Translate [**59**], it responds with "un sonido cálido." While the word-for-word translation is correct, the appropriate translation to correctly express the meaning is "un sonido profundo." The word-for-word translation to English of "un sonido profundo" is "a deep sound," not "a warm sound."

As a result, people relying on current translation technology may fail to communicate while believing they have. This, for example, would make it difficult for an English-speaking audiologist to correctly diagnose hearing problems for people whose primary language is not English. It would also make it difficult for two musicians to communicate to each other during a recording or mixing session—requiring them to rely on other means of communication besides descriptive language. If we develop semantic audio tools that understand descriptive language, we cannot ignore the fact that English is not the first language of many users. To maintain interface consistency when internationalizing audio software, we must learn how to translate in the audio sense from one language to another.

In this section, we address the question, "What equalization descriptors are true audio synonyms *between languages*?", and we describe how we can use SocialEQ to build a translation map between sound adjectives of two languages: English and Spanish. Using the same method we used to calculate audio descriptor synonyms in Section 2.5.2.5, we translate between languages. When the two words are synonyms but come from different languages, we consider one a translation of the other. The more frequently a pairing between two words occurs, the more certain the translation.

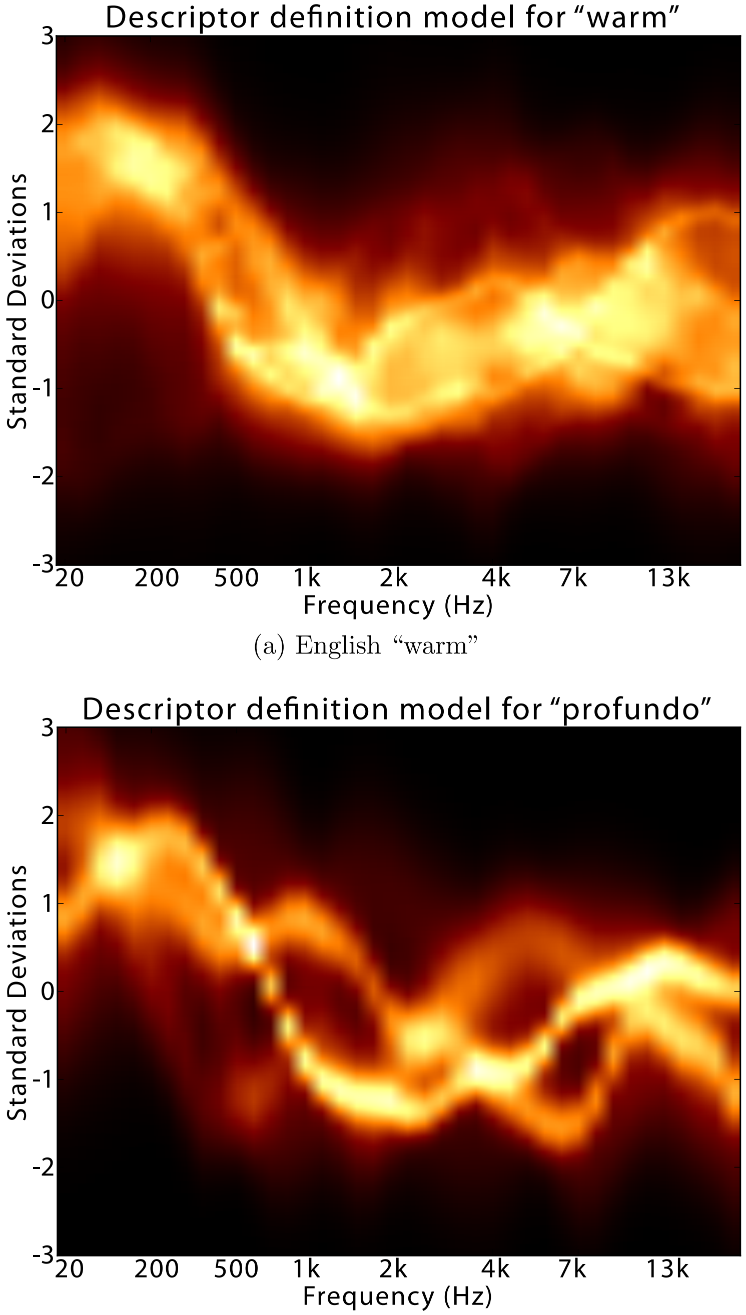
2.6.1. Additional Data Collection

To build a collection of audio descriptors in Spanish, we again recruited participants through AMT as we described in Section 2.5.1. However, the AMT Human Intelligence Task (HIT)'s title and description were in Spanish this time. After combining the English and Spanish data, we had 887 participants who participated in a total of 2322 training sessions (one session per learned word). Of the 2322 training sessions, 983 of them were contributed by participants recruited in English and used a version of the SocialEQ system in which all of the instructions were in English. The other 1339 sessions were contributed by participants recruited in Spanish who used a version of SocialEQ with instructions in Spanish.

We used the same inclusion criteria as specified in Section 2.5.1.1 to remove contributions from inconsistent participants (repeated audio examples were labeled very differently) and those who showed no effort (e.g., completed labeling 40 sounds in under 1 minute). This left 676 participants who taught the system in 1602 sessions. Of these sessions, 923 were English and 679 were Spanish, resulting in 388 unique English and 384 unique Spanish descriptive terms. The median number of descriptive terms contributed per participant was 1. Table 2.6 shows the top 10 descriptive terms in each language ranked by the *agreement score* defined in Section 2.5.2.4.

2.6.2. Mapping Between English and Spanish

To determine the relationships of the high-agreement words from Table 2.6, we performed agglomerative hierarchical clustering as we did in Section 2.5.2.5 and plotted the dendrogram in Figure 2.14. From this plot, we can see that the models displayed in Figure 2.13



(a) English "warm"

(b) Spanish "profundo"

Figure 2.13. Two closely related descriptor definition models: "warm" (N=64) and "profundo" (N=6), where N indicates how many people trained the system on the word in question.

Rank	English word	Sessions	Agreement Score
1	tinny	8	0.294
2	quiet	5	0.188
3	deep	6	0.164
4	light	6	0.151
5	warm	64	0.139
6	loud	26	0.137
7	heavy	15	0.124
8	dark	8	0.122
9	bright	19	0.112
10	energetic	5	0.107

Rank	Spanish word	Sessions	Agreement Score
1	pesado	10	0.142
2	agudo	5	0.111
3	suave	23	0.100
4	bajo	5	0.094
5	claro	33	0.092
6	fuerte	18	0.083
7	dulce	10	0.080
8	tranquilo	13	0.068
9	profundo	6	0.065
10	frío	13	0.063

Table 2.6. Top 10 English and Spanish equalization descriptors ranked by *agreement score*

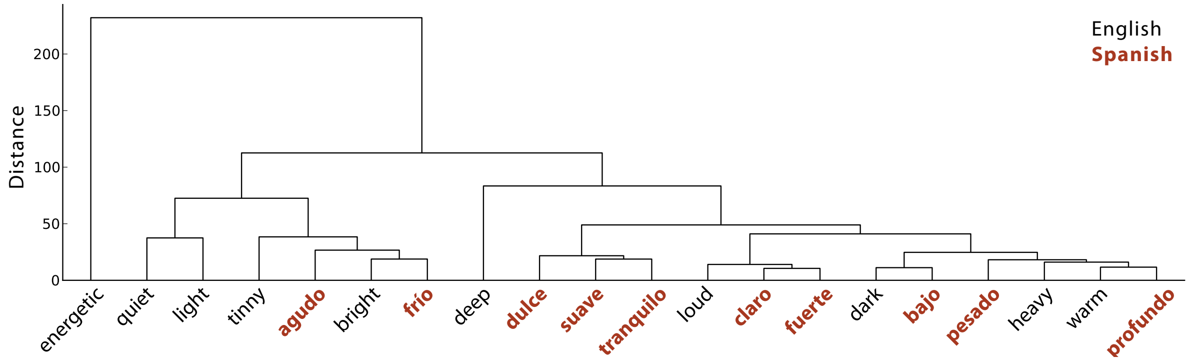


Figure 2.14. Hierarchical clustering of the 10 descriptors with the highest agreement scores in each language.

(“warm” and “profundo”) are closely related despite that “warm” typically translates to “cálido” in Spanish.

Training SocialEQ with multiple languages provides an alternative to typical dictionary-based and statistical machine translation. This method of translating by collectively teaching intermediate abstract representations can potentially be extended to other domains, uncovering unknown relationships between languages.

2.7. Using an Audio Descriptor Map for Control

In this section, I am going to briefly present work by Prem Seetharaman and Bryan Pardo that builds on the work of SocialEQ and demonstrates SocialEQ's effectiveness at building a two-dimensional descriptor map interface.

Audealize is an audio production interface by Seetharaman and Pardo [**167**] that embodies and extends the ideas and data of SocialEQ. While SocialEQ outlined how to crowdsource the construction of an audio-equalization descriptor map, Audealize implements novice-taught two-dimensional descriptor maps for both equalization (from SocialEQ) and reverberation (from SocialReverb [**166**]—follow on work to SocialEQ by Seetharaman and Pardo that crowdsourced reverberation descriptors) into an interface that lets the user modify the sound by selecting descriptive terms (e.g., "underwater", "tinny", "chaotic") on the map. In SocialEQ, we model an audio-equalization concept as a probability distribution rather than a single parameter setting. To convert these representations into a parameter setting, Seetharaman and Pardo took the mean of the audio-equalization concept models to obtain parameter weights. Once they had the weights, they multiplied them by a user-specified gain to obtain a parameter setting.

In a within-subject user study of the system, they compared the map-based interface to a traditional graphic equalizer interface. There were two different kinds of tasks: a *match word* task, in which the participant had to use the tools to achieve a particular semantic goal (e.g., "tinny"), and a *match effect* task, in which the participant had to match the sound of a particular audio example that is achievable by the processing tool. In a survey which assessed user satisfaction with the interfaces, they found that the map-based equalization tool performed better than the traditional interface for both *match word* and *match effect* on all measures—mean user response of agreement (ranging from 0 (*strongly disagree*) to 1 (*strongly agree*)) for the following statements: "I achieved my goal", "I was satisfied with my experience using this interface", "I was able to find relevant audio effects easily", "I enjoyed using this interface", "I understood how to use this interface to achieve a specific goal". For the *match effect* task, they also measured the Pearson correlation between the parameters of the target effect and the parameters the participant set. Using this performance measure, they also found the map-based interface performed better (i.e., higher correlation) than the traditional interface. This study validates the utility of a novice-taught two-dimensional descriptor map interface for audio production, and also validates the utility of the SocialEQ data.

2.8. Post-Research Related Work

Since SocialEQ was submitted for publication, a number of related works have also been published that build semantic and/or crowdsourced design tools in various mediums.

As mentioned in Section 2.7, Seetharaman and Pardo followed on SocialEQ with SocialReverb [**166**], a project to learn audio reverberation concepts from novices, and by combining the data from SocialEQ and SocialReverb they created a two-dimensional descriptor map interface called Audealize [**167**]. Kim and Pardo [**93**] also followed on the SocialEQ work by speeding up the user-concept learning process using a collaborative filtering technique that is more flexible than the earlier collaborative filtering work by Pardo et al [**134, 133**].

Stables et al [**178**] took a different approach to semantic audio processing. Instead of creating a web-based system and using a crowdsourcing platform to recruit participants to teach the system, they created SAFE, a set of audio plugins (an equalizer, reverberator, compressor, overdrive) in which users contribute descriptive terms while working in a typical Digital Audio Workstation (DAW) environment. The tools use traditional interfaces (e.g., similar to Figure 2.2 for the equalizer), but they have an additional widget for contributing descriptors or recalling previously contributed descriptors. In addition to the descriptor and processing parameters, their system also collects the features of the input and output signal. This approach has the benefit of collecting descriptive terms in the typical environment in which it would be used, but it has the detriment of requiring the contributors to use the traditional interface, limiting its use to the vocabulary of experts. In a subsequent analysis of their data, Stables et al [**177**] investigated the relationship of descriptors used within a processing type (e.g., equalization) and between processing types (e.g., equalization vs. reverberation), and they discovered insights such as that equalization and distortion share a common vocabulary, while distortion and reverb have very dissimilar vocabularies. In [**179, 180**], the researchers extended the SAFE interface with a two-dimensional descriptor map. The map was built using an auto encoder to map a weighted parameter space down to two dimensions and was trained using 800 contributions from 40 participants equalizing 10 musical instrument samples with two descriptors: “warm” and “bright”.

With similar motivations as Sabin et al [**156**], Huang et al presented a method to learn personalized semantic controls for synthesizers. They used an evaluation-based approach similar to the work of Sabin et al. However, synthesizers typically have a more complex mapping from the parameter space to the perceptual space. Therefore, they employed a

Gaussian process approach with active learning since it can learn more complex functions than the linear regression-based approach of Sabin et al.

Researchers have also presented related work outside of the audio domain. In Attribit [**27**], Chaudhuri et al present a method for crowdsourcing the learning of semantic attributes (i.e. descriptive terms) for 3D models and web pages, as well as an interface for generating 3D models from exisiting parts given semantic attributes. Their method employs a rank learning algorithm trained on pairwise comparisons collected from AMT workers. In their optimization function they use the aggregate opinion of three annotators for the pairwise judgments, and they weight the aggregate judgment based on agreement. In related work [**201**] published a couple of years later, Yumer et al presented a method to crowdsource the learning of another 3D modeling interface controlled by semantic attributes, but this interface deforms existing models to achieve a goal rather than constructing new ones from exisiting parts (as in Attribit). In addition 3D modeling, domains such as font design [**128**] and image processing [**99**] have also utilized interfaces controlled by crowdsourced semantic attributes. The diverse interest in this type of interaction in many design domains attests to the potential generality and utility of this type of interaction.

2.9. Conclusion

In this chapter, I presented my work on communicating audio concepts to software with descriptive language. I proposed a novel method for learning actionable "dictionaries" of audio-descriptors for audio-production tools—i.e., maps from audio descriptors to audio concepts. This method allows a population of users to contribute audio-descriptors by simply listening to and evaluating audio examples. The method aggregates these contributions into audio-descriptor definitions. The descriptive terms that are indexed in this dictionary aren't simply described by other words, but by perceptually-relevant abstract representation that can be used to control audio-production tools. Using this method, I collected and released the first large-scale data collection of audio descriptors for audio production tools. Using this data, I showed how we can use these representations to find agreed upon meanings, compare definitions to each other to find synonyms, build two-dimensional descriptor maps, and translate audio descriptors between languages. Audio production tools can use these learned dictionaries to provide affordances for novices, enabling them to produce their desired audio-concept using descriptive language rather than difficult-to-understand, low-level technical parameters.

While I used this method to collect descriptors for audio equalization, this method can also be applied to other audio production and other creative media production tools. In addition, if we use this method to collect data from several media production tools in isolation and combination, we can begin to learn how descriptors map on to several tools at once. For example, distortion and/or reverberation may be needed in addition to equalization for some user's definition of "warm". However, more complicated audio production tools (e.g., synthesizers) require changes to the modeling technique. For example, learning audio concepts for tools with dependent parameters or parameters that are perceptually non-linear, may require replacing the current user-concept learning mechanism (by Sabin et al [**156**]) with the more recently introduced method by Huang [**77**].

Lastly, mapping the descriptor definition models to the parameter space of an audio production tool may be problematic if the models are multimodal and/or there is little agreement between the contributors on the meaning of an audio descriptor. In the next chapter, I discuss a solution to this problem.

CHAPTER 3

Evaluative Feedback: Crowdsourced Audio Quality Evaluation

3.1. Overview

In this chapter, I present my work on communicating audio concepts to software using evaluative feedback. However, rather than focusing on how an *individual* user can communicate an audio concept to audio production software using evaluative feedback, I'm going to present work on how a *population* of listeners can communicate general audio concepts to software. As in SocialEQ, this can be used to enable higher-level communication methods such as descriptive language or exploratory maps that provide affordances in audio production tools to novices.

In Chapter 2, I presented a method that used simple evaluative interactions to crowdsource the learning of a semantic mapping that can be used to provide affordances in audio production tools for novices. To do so, I asked a crowdsourced population of novices to describe audio descriptors by simply *evaluating* audio examples—a task that any listener can do. From each individual's set of audio-example ratings, I built a personal audio concept model, and I then constructed descriptor definition by aggregating the learned personal audio concept models from several individuals into a single, definitive model. However, the method I used to aggregate models has its limitations (discussed below).

In this chapter, I propose alternative approaches to aggregating crowdsourced evaluation data from individuals that overcome these limitations; rather than aggregating learned models, I instead aggregate the audio evaluations, which we then can subsequently use to build a single definitive model. In addition to aggregating data for training models, these methods also enable researchers to quickly and easily run crowdsourced listening tests for evaluating new audio algorithms. It's through the lens of crowdsourced listening tests that I present this research.

The work described in this chapter was presented at the International Conference on Acoustics, Speech and Signal Processing [**24**] and has been recently submitted for publication in a journal.

3.2. Introduction

3.2.1. Evaluating audio quality for generating training data agreed upon by a population

When building the dictionary of audio descriptors in SocialEQ, the system learned many individual's audio concept models which we then aggregated into definitive models. By aggregating individuals' audio concepts of a particular descriptor, we estimate how a population as a whole defines a particular audio concept. Using these models, we created a semantic mapping that can be used to provide affordances in audio production tools to novices through the use of descriptive language (i.e. audio descriptors). Unfortunately, there are limitations to the SocialEQ approach of data aggregation.

First, SocialEQ's method for aggregating audio concept models into a definitive model produces a rich, flexible representation for comparing audio-equalization concepts, but the representation can be problematic when mapping certain audio-concepts to low-level parameter settings. For example, the audio concept models that we learned are probability distributions over a frequency-weighting space. This representation works well for defining audio-equalization concepts and calculating similarity, but in order to use these models to map from the semantic audio concept space to the parameter space of an audio equalizer, we must map the distribution in the frequency-weighting space down to single points in the parameter space. In Audealize [**167**], which was partially built on the SocialEQ data, the authors did this by simply taking the mean of the audio concept models to obtain parameter weights. Once they had the weights, they multiplied them by a user-specified gain to obtain a parameter setting. This approach is adequate if the probability distributions in the model are unimodal, but it is problematic if the distributions are multimodal and/or there is little agreement between the contributors on the meaning of an audio-descriptor.

Second, contributors to SocialEQ may have very different hearing abilities and be in very different listening environments. The frequency responses of individuals' hearing and environments can affect their evaluations and therefore the learned models. For example, imagine someone is contributing "brittle" to the SocialEQ system, but they are listening over laptop speakers that cannot reproduce frequencies less than 200Hz. In that scenario, the evaluator may positively rate audio examples with strong low-frequency components even if they consider "brittle" as lacking in low-frequency content.

Third, it can be difficult for contributors to rate consistently due to the high number of required ratings. Each contributor has to rate 40 audio examples on a continuous scale. 15 of these audio examples are repeated so that we can calculate a consistency value that is used to scale the learned model during aggregation. However, each new example must be rated in relation to all previous examples, which makes it difficult to maintain consistency. The Multi Stimulus test with Hidden Reference and Anchor (MUSHRA) listening test recommendation addresses this by limiting each evaluation to only 12 audio examples.

In this chapter, I explore alternative approaches to aggregating crowdsourced audio evaluations that overcome these limitations. Rather than learning a different model from the audio evaluations of each contributor and then aggregating the models, we instead aggregate the audio evaluations themselves, which we then can subsequently use to build a single definitive model. In addition, by aggregating the audio evaluations themselves, the aggregate audio evaluations can be used to train classification and regression models as well. I also present methods that can account for varied hearing responses and listening environments and that can also support more than 12 audio examples while still producing discriminative results.

3.2.2. Evaluating audio quality for comparing audio systems and algorithms

I approach this problem of crowdsourcing and aggregating audio evaluations through the lens of lab-based audio evaluation listening tests such as MUSHRA [**82**], which are used to produce “gold-standard” audio evaluation data. There are several advantages of approaching this problem from the perspective of listening tests. Researchers have extensively studied standardized listening tests such as MUSHRA in lab environments and have optimized them to obtain discriminative evaluation scores from a population of listeners. We can therefore build on their testing methodology and test, adapt, and extend them for crowdsourced scenarios. Because of their popularity, data from lab-based listening tests is readily available to which we can compare web-based (e.g., crowdsourced) listening tests. Lastly, in addition to obtaining and aggregating data to train to models, a recommended methodology for crowdsourced listening tests would make performing listening tests fast and easy for researchers—an important contribution on its own.

For example, a goal of many researchers in audio processing and synthesis is to create algorithms that produce output that “sounds good” to human listeners. The gold-standard evaluation measure for good-sounding audio is a lab-based listening test, such as the MUSHRA recommendation [**82**]. Conducting such a study is often time consuming and difficult. Therefore, in place of listening tests, researchers frequently use automated measures of signal quality that have limited correlation to human quality estimates and typically require ground truth data for comparison. If researchers could run listening tests more easily and quickly, they would be more likely to evaluate how good their algorithms actually sound to human audiences. If researchers could run these tests without reference to ground truth, they could furthermore evaluate how well their algorithms perform on wider and more representative sets of real-world examples.

I use audio source separation evaluation as our example task in our study of crowdsourced audio quality evaluation, but I believe that the results from this study should generalize to any audio evaluation task in which the differences in audio examples are easily perceptible. Audio source separation is a task that aims to extract a clean recording of one or more target sounds (e.g., a vocalist or a choir of vocalists) from a recording containing a mixture of several sounds (e.g., the rest of a band) [**195**]. This is a task that has received significant interest in the last few decades. It is representative of tasks for which the algorithms’ output have easily perceptible artifacts that would be considered medium or large impairments according the International Telecommunication Union

(ITU) recommendations [**82, 81**]. Because the artifacts are easily perceptible, the difficulty in evaluating algorithms for such tasks is not in detecting the artifacts but rather assessing the relative annoyance of various artifacts [**82**].

There are two common automated objective methods for the evaluation of audio source separation algorithms: BSS-Eval [**193**] and Perceptual Evaluation methods for Audio Source Separation (PEASS) [**42**]. Unfortunately, these measures do not correlate well with human ratings of audio quality (e.g., Pearson $r \sim 0.50$ for overall quality measure for BSS-Eval) [**42**] and they require ground truth separated sources. PEASS is also dependent on the artifacts in the audio it was trained on, causing its output to be unpredictable on more recent audio source separation algorithms. The alternative to automated evaluation methods is asking a population of listeners (typically experts) to compare algorithms' output to each other and/or to signals with agreed upon quality on one or more dimensions in a subjective listening test.

Subjective listening tests, such as MUSHRA [**82**], can easily be modified to assess the quality of different audio tasks, and some subjective listening tests (e.g., mean opinion or pairwise-comparison) can be administered even when no ground truth exists. Subjective listening tests typically require recruiting and administering tests to at least 20 experimental participants in a lab and can easily consume a week of a researcher's time.

Researchers have sought to reduce the time and effort to run these tests by moving subjective listening tests from the lab to the web [**95, 148, 161, 162, 28, 87**]. This allows researchers to recruit participants via crowdsourcing labor markets such as Amazon's Mechanical Turk (AMT) or citizen scientist projects such as LabintheWild [**146**].

Crowdsourcing platforms such as AMT and LabintheWild automate recruiting participants and running tests, enabling a researcher to run tests in hours instead of days or weeks, with much less effort.

However, moving from the lab to a web-based crowdsourcing platform introduces variability: varied reliability of assessors, levels of expertise, listening environments, listening devices, and hearing abilities. Researchers have assessed crowdsourced media quality evaluations before [**48, 159, 92, 28, 148**], but neither those researchers nor the developers of browser-based MUSHRA frameworks [**95, 162, 87**] compared the results of a crowdsourced listening test on the web to a gold-standard MUSHRA listening test conducted in a lab. Therefore, it was unclear how this variability affects gold-standard listening tests such as MUSHRA.

3.2.3. Overview of our approach

To investigate the crowdsourcing of audio quality evaluations, I evaluate two types of web-based listening tests: a web-based MUSHRA-like[1] test (which I will refer to as *web-MUSHRA*) and a web-based pairwise-comparison test. Both of the web-based tests are performed in an uncontrolled web environment on a population drawn from AMT. I also propose two methods for gathering information about participants' varied listening environments and hearing abilities, and I incorporate this information into our score estimation procedures. I compare both web-MUSHRA and the pairwise-comparison tests

[1] Some of the specifications of the MUSHRA recommendation are not feasible on the web (e.g., playback system specifications and the requirement for expert users). Therefore, I refer to such tests performed on the web as "MUSHRA-like" tests—as they are multi-stimulus tests that still share enough similarities with MUSHRA that it is still a useful point of reference.

to MUSHRA performed in a controlled lab environment (which I will refer to as *lab-MUSHRA*).

I chose to adapt MUSHRA for a crowdsourced web-based environment due to the test's popularity for evaluation in lab-based environment. MUSHRA has some limitations though that make it unsuitable for some audio evaluation scenarios: it requires "ground-truth" audio signals for reference, and it is limited to evaluating at most 12 audio examples (i.e. *stimuli*) at once. MUSHRA's and the automated evaluation measures' (PEASS and BSS-Eval) reliance on ground truth data is a major limitation as it limits evaluating audio source separation algorithms on data sets that have separated sources available, which are scarce and often unrealistic.

Pairwise-comparison listening tests, however, do not have these limitations. They can estimate quality scores with or without ground truth stimuli for reference. This allows evaluation to be performed on any kind of real-world data. Also, there is no limit on the number of stimuli that can be evaluated. When an additional stimulus is compared, individual tasks are not more taxing on the participant—we need more pairwise comparisons per trial, but these comparisons can be distributed among multiple participants. Additionally, techniques have recently been developed to further reduce the number of comparisons [**143, 129, 29**].

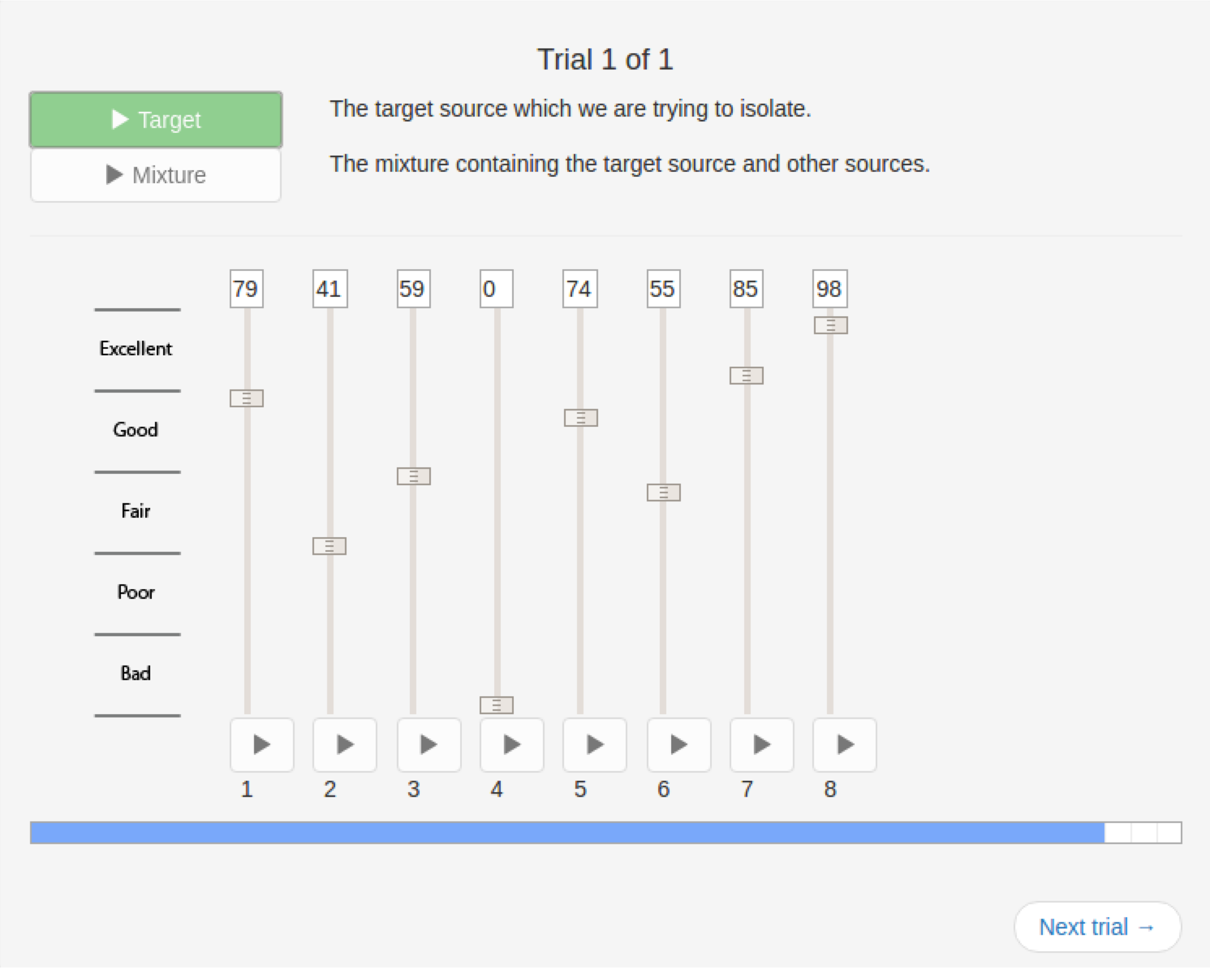
Pairwise-comparison listening tests may also have other advantages over MUSHRA in a crowdsourced setting. Pairwise-comparison tests are discriminative by design and require participants to attend to fewer stimuli simultaneously than MUSHRA—a characteristic that could be beneficial using novice, crowdsourced participants. In pairwise-comparison tests, we can also determine participant reliability by simply observing transitivity in their comparisons [**76, 28**], whereas MUSHRA requires repeated trials to determine a participant's reliability. For example, if a participant chooses stimulus A over B, and B over C, then by transitivity we expect them to choose A over C as well—if they don't, then they have violated transitivity. By observing the frequency of transitivity violations, we can measure a participant's consistency. A measure of consistency is not only useful for incentivizing workers with consistency-based rewards, but may also be useful knowledge when aggregating ratings. Therefore, pairwise-comparison tests may be more suitable for large-scale, crowdsourced quality evaluations.

Therefore, I compare both web-based MUSHRA and pairwise-comparison listening tests using a population recruited from AMT and compare the resulting aggregate quality scores to "gold-standard" lab-based MUSHRA scores.

3.3. Background

3.3.1. Overview of MUSHRA

The MUSHRA recommendation describes a multi-stimulus listening test for the subjective assessment of intermediate audio quality. In a single MUSHRA trial (see Table 3.1 for a definition of terms), up to 12 stimuli are rated in comparison to a reference and each other on a 0–100 continuous quality scale using a set of sliders (see Figure 3.1). A participant can play the stimuli as many times as they want. Each time a stimulus is played, it is played from the beginning. One of these stimuli is the hidden reference (the desired sound, not identified by a label) and at least one other is an unlabeled low anchor (typically a very "bad" sound). The remaining stimuli are outputs from the systems under test. Since these stimuli are rated in comparison to the reference, we expect the reference to



Trial 1 of 1

The target source which we are trying to isolate.

The mixture containing the target source and other sources.

79 41 59 0 74 55 85 98

1 2 3 4 5 6 7 8

Figure 3.1. The interface for the MUSHRA listening test running on the Crowdsourced Audio Quality Evaluation (CAQE) software.

be rated as excellent. Anchors are stimuli designed to be rated as poor. The stimuli and the non-hidden reference can be played and rated unlimited times in each trial before the ratings are submitted. The recommendation also specifies the listening environment, training procedure, participant screening, and participant selection (participants must be experienced, normal-hearing listeners, properly trained in subjective quality evaluation).

Table 3.1. Definitions of terms

lab-MUSHRA	The lab-based MUSHRA listening test that generated the PEASS data set.
web-MUSHRA	The crowdsourced web-based MUSHRA-like listening test presented in this paper.
mixture	In audio source separation, an audio mixture is a signal composed of the summed signals of several audio sources. In the PEASS data set, there were 10 different mixtures.
target	In audio source separation, the target is the audio source which we want to separate from the mixture. Each mixture in the PEASS data set has a different target.
quality scale	A scale on which an audio stimulus is evaluated. In the PEASS data set there were 4 different quality scales.
condition	A testing condition. There were 40 conditions in the PEASS data set, the Cartesian product of the set of mixtures and the set of quality scales.
score	A quality score for a stimulus estimated by either a subjective (e.g., MUSHRA, pairwise-comparison listening test) or an objective (e.g., BSS-Eval, PEASS) testing method.
trial	A single experimental instance of a particular testing condition completed by one participant.
HIT	Human Intelligence Task, a unit of work on Amazon's Mechanical Turk.
reference	The reference audio stimulus to which the other audio stimuli should be compared in a listening test. In audio source separation evaluation, the reference is the target source. In MUSHRA, there is always a "hidden" reference as well, which should always be given a high rating.
anchor	A audio stimulus which has been constructed to have an expected rating (a low rating in this work) on a particular quality scale. Each quality scale in a MUSHRA listening test should have at least one low anchor stimulus.

3.3.2. Automated Source Separation Quality Measures

Audio source separation is a task that aims to extract a clean recording of a single target sound (e.g., a vocalist) from a recording containing a mixture of several sounds (e.g., the rest of a band) [**195**]. There are two common methods for objective evaluation of audio source separation algorithms: BSS-Eval [**193**] and PEASS [**42**]. The BSS-Eval method decomposes the energy of the separated signal by projecting the separated signal onto the target source and interference sources and extracting the residual. It then uses this decomposed energy to construct signal measures that consist of energy ratios: source-to-distortion ratio (SDR), image-to-spatial-distortion ratio (ISR), source-to-interference ratio (SIR), and sources-to-artifacts ratio (SAR). Unfortunately, these measures do not correlate well with human ratings of audio quality (Pearson $r \sim 0.50$ for overall quality measure) [**42**].

This poor correlation is because some perceived differences in signals can be more annoying than others, despite being lower in energy. Suppose, for example, that the output of a audio source separation algorithm that has significantly reduced the volume of non-target sounds but not completely eliminated them. This will likely be rated as higher quality than the output of an algorithm that has completely eliminated the non-target sources but has introduced annoying processing artifacts (e.g., 'musical'/spectral noise that sounds like random 'bleeps'). However, these two outputs may have the same SDR.

To address this discrepancy, researchers developed objective quality measures for audio source separation (e.g., PEASS) based on models of human audition and trained on human-labeled data [**42, 50**]. PEASS performs better than BSS-Eval on the data it was trained on. However, while the developers of PEASS used a diverse set of audio source separation algorithms from the 2008 Signal Separation Evaluation Campaign (SISEC) [**192**], many new types of audio source separation algorithms have been developed since then. New types of algorithms have new types of artifacts that PEASS wasn't trained on.

PEASS's performance on these algorithms is unpredictable. To overcome these drawbacks, researchers must evaluate algorithms with listening tests.

3.3.3. Baseline Data Set

To train the PEASS objective scoring models, the developers of PEASS, Emiya et al., conducted lab-based listening tests [**42**]. We use the same test audio as these lab-based listening tests and treat the lab results as the gold-standard baseline.

This test material consists of 10 mixtures (5 speech, 5 music) which are each 5 seconds long. For each mixture, there are 8 test sounds: the ground-truth target source (the reference), 3 anchors, and 4 outputs of a variety of audio source separation algorithms from the 2008 Signal Separation Evaluation Campaign [**192**]. This data set was collected in a lab setting according to the 2003 MUSHRA recommendation ITU-R BS.1534-1 [**80**]. There were 20 normal-hearing participants who were experts in general audio applications. Each participant was consented via a script and performed a MUSHRA trial for 4 different quality scales (not randomized) for each of the 10 sets of test sounds (randomized) while listening over the same model of headphones in a quiet environment. The quality scales were in terms of *global quality*, *preservation of the target source*, *suppression of other sources*, and *absence of artificial noises*. The three anchors were constructed so that all three should be rated low for the *global quality* scale, but only one should be rated low for each of the remaining quality scales. In total there were 7360 ($23 \times 10 \times 8 \times 4$) ratings. We refer to this set of lab-based MUSHRA ratings as the PEASS data set.

3.4. Crowdsourced Web-based MUSHRA-like Listening Tests

We first establish that the qualitative results from a lab-based MUSHRA test can be duplicated using crowdsourcing over the web.

3.4.1. Methods

To perform our MUSHRA-like listening test on the web, we utilized Amazon's Mechanical Turk to recruit and pay participants. In the original PEASS data collection, participants performed 40 different MUSHRA trials, one for each mixture / quality-scale pair. This took each participant several hours to complete. On AMT, however, a Human Intelligence Task (HIT) is expected to take only a few minutes or less [**131**], and it is recommended to keep media quality evaluation tasks less than a minute if possible [**75**]. We therefore limited each HIT to be one MUSHRA trial. A MUSHRA trial typically takes a few minutes to complete, but it is the smallest possible unit of work for this testing methodology. We assigned participants to one of the four quality scales, and allowed each participant to perform up to 10 MUSHRA trials, one for each mixture (randomly ordered). We limited each participant to one quality scale to reduce task confusion and to eliminate any quality scale ordering effects.

The MUSHRA recommendation specifies that all participants must listen in similar, controlled listening environments. However, when running tests via AMT, the participants will be in a large variety of environments. To account for this, we asked participants to listen over headphones and to perform two different hearing tests in addition to the MUSHRA trial. Participants only had to perform these tests once, regardless of the number of MUSHRA trials they did.

The first hearing test—the *hearing screening*—ensured participants listened over a device with an adequate frequency response (e.g., *not* laptop speakers) and followed instructions. This test was administered after participants accepted the HIT on AMT. In this test, participants were first asked to adjust the volume of a 1000Hz sine tone to a comfortable level and to not change the level thereafter. Participants listened to two 8s audio clips and counted how many tones were heard. Each clip contained a 55Hz tone, a 10kHz tone, and between 0 and 6 tones of random frequencies between 55Hz and 10kHz. Tones were 750ms sine waves spaced by 250ms of silence. Tones were scaled to be of approximately equal loudness on speakers with a flat frequency response at a comfortable playback level, and they were presented in random order with silence replacing tones when there were less than 8 tones. Participants had two chances to answer correctly. However, if they did not answer correctly, we allowed them to proceed as if they did; this allows us to test the effect of this screening process.

The second hearing test—the *in situ hearing response estimation*—obtained an overall estimate of hearing thresholds at a range of frequencies. This hearing threshold is a combination of the frequency response of the environment, their hearing, and their listening device. This test was administered after the first MUSHRA trial. Participants again listened to audio clips and counted tones of the same duration as the hearing screening. There were eight 12s audio clips in this test. One clip contained only silence. In the remaining clips, the frequency of the sine tones was constant throughout the clip, but the amplitude of the tones varied. The seven frequencies were log-spaced between 23Hz and 16.8kHz. Each clip contained randomly-ordered tones of six 15dBFS-spaced levels (-90 to -15dBFS) and up to 3 additional repeated tones. The remaining time consisted of silent beats. Ordering of tones and beats was random. Based on a participant's tone count for a particular frequency, we determined their in situ hearing threshold at that frequency. Since a participant may not pay attention or may answer randomly, we established an inclusion criteria that was satisfied if a participant's tone count was not zero for all frequencies and was not more than one higher than the actual number of tones. If this criteria was not satisfied, their response was saved but marked as rejected.

While the *in situ hearing response estimation* is a longer listening test that essentially contains the frequency response information of the *hearing screening*, it does not contain all of the test's information. The *hearing screening* is an objective test that has a correct answer, while the *hearing response estimation* does not. Therefore, the *hearing screening* tells us not only if a participant listened over a device with adequate frequency response but also if that participant is cooperative and paying attention.

Lastly, we discovered in a pilot study that some language used in the PEASS MUSHRA instructions was inappropriate for novice participants. To reduce participant confusion, we simplified the instructions from the instructions used in the PEASS data collection by rephrasing and rewording the instructions to be as clear as possible to a novice unfamiliar with audio source separation. We also added an additional training step for participants in which we played example anchors and references (that were not used in the rest of the study) and informed them of the clips' expected ratings.

3.4.2. Data Collection

We collected *a minimum of* 20 MUSHRA trials for each condition (mixture / quality scale pair) by participants that passed the hearing screening. Note that this typically

Table 3.2. Trial statistics of web-MUSHRA listening test

Listening test type	**web-MUSHRA**
No. of participants	530
No. of participants that passed hearing screening	336
Participant reported gender	
- No. female	184
- No. male	346
Participant reported age	
- min	18
- max	67
- mean	31.0
- median	28
No. of Trials	1763
Trials per condition	
- min	26
- max	55
- mean	34.4
No. of trials by participants that passed hearing screening	1147
Trials per condition by participants that passed hearing screening	
- min	20
- max	37
- mean	23.6
Trials per participant	
- min	1
- max	10
- mean	3.3

meant we collected a larger number of trials per condition, but that not all participants in a trial passed the screening. We paid participants $0.80 for completing an initial trial, which included the hearing tests, and $0.50 for subsequent trials. Only AMT workers who had at least 1000 AMT assignments approved and a 97% approval rate were allowed to participate. Trial and participant statistics are shown in Table 3.2.

We also performed a participant survey. According to this survey, the distribution of reported listening devices was 72% headphones, 16% laptop speakers, 10% loudspeakers.

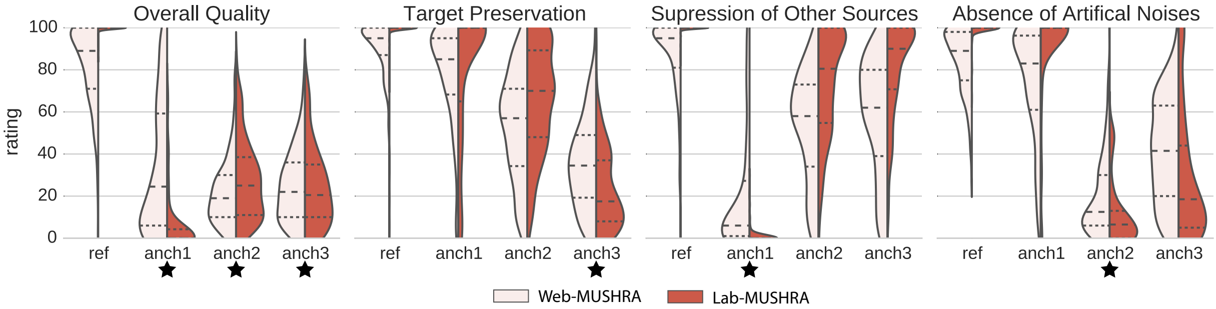


Figure 3.2. Distribution of reference (ref) and anchor ratings (anch1–3) for the 4 quality scales pooled over all mixtures in web-MUSHRA and lab-MUSHRA. The stars below the anchors indicate which anchors are expected to be rated low for that quality scale. The dotted lines are the quartile and median markings. Anchor 1: the sum of all sources (the mixture), Anchor 2: target + ‘spectral noise artifacts’, Anchor 3: low-passed target with 20% of time-domain frames missing (see [**42**] for more details)

.

After the hearing screening, this distribution changed to: 84% headphones, 3% laptop speakers, 11% loudspeakers. In addition, 44% of survey respondents reported being able to hear non-test sounds (e.g., environmental sounds) during the test, but only 7% of respondents found these sounds distracting. 85% of participants reported to never having participated in a listening-based study before. While expected, this confirms that this population is primarily non-experts in a variety of uncontrolled listening environments.

3.4.3. Results

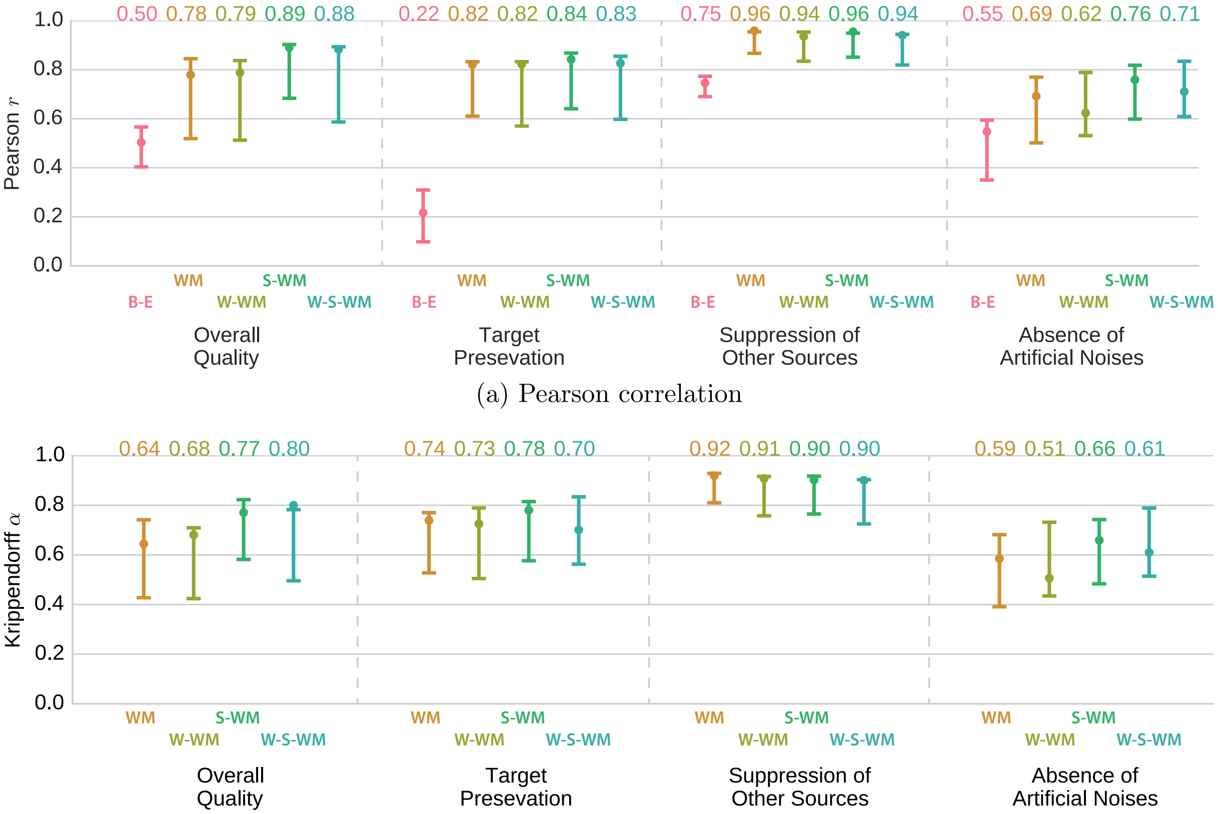
3.4.3.1. Did the participants understand the task and rate reasonably? We expect participants who understood the task and quality scale to rate the quality scale’s anchor(s) (indicated by the asterisks in Figure 3.2) in the lower half of the scale, and the other quality scales’ anchors in the upper half of the scale. We also expect references to be rated very high, near 100.

Figure 3.2 shows distributions of participant ratings of the reference and anchor sounds from the PEASS data. Each sub-figure corresponds to a quality scale (e.g., absence of artificial noises). Within a quality scale there are four violin plots. The left half of each plot gives the distribution of web participants, the right half gives the distribution of lab participants. A star below a plot indicates that stimulus should be rated low if the participant understands the task.

In general, both web and lab participant distributions of ratings indicate they understand the task, as distributions skew low for stimuli marked by an asterisk and high for the others. The main exception to this is on the *absence of artificial noise* quality-scale, the distribution of *anch3* is centered on the wrong side of the scale for both the lab and web participants. While the use of training examples mitigated this effect for the web participants, the median for the web-MUSHRA's *absence of artificial noise anch3* is still only 42, when it should be above 50 since it is not an anchor for that scale—it's an anchor for *target preservation*. It seems that listeners hear any distortion (additive or subtractive) to the target as simply a distortion. This conflation makes the *target preservation* and *absence of artificial noise* quality scales problematic. We believe using a single quality scale that is inclusive of all distortions (e.g., *lack of distortions to the target*) would eliminate this confusion.

3.4.3.2. How similar are the web-MUSHRA scores to lab-MUSHRA scores?

To answer this question and establish if an AMT-based listening test can act as a proxy to a lab-based test, we measured the Pearson correlation between web-MUSHRA scores with lab-MUSHRA scores. As recommended in the MUSHRA standard, we used the median to aggregate participants' ratings into stimulus scores. Figure 3.3a displays 95%



(a) Pearson correlation

(b) Krippendorff's alpha agreement

Figure 3.3. Pearson correlation and Krippendorff's alpha agreement of web-MUSHRA and BSS-Eval scores with the lab-MUSHRA scores for the 4 quality scales. Scores were limited to the systems under test (i.e. excluding the reference and anchors) and estimated using the median of ratings from a sample size of 20 participants per mixture. Scores for all mixtures were concatenated before calculating the correlation for each quality scale ($N = 40$). Bars represent 95% confidence intervals (CIs) calculated from 1000 bootstrap iterations, randomly sampling with replacement from each sample group. **B-E**='BSS-Eval', **WM**='Web-MUSHRA', **W-WM**='Weighted Web-MUSHRA', **S-WM**='Screened Web-MUSHRA', **W-S-WM**='Weighted and Screened Web-MUSHRA'

confidence intervals and point estimates of the correlation. This figure also contains the correlations of other web-MUSHRA variations (*weighted* and *screened*) which we will discuss below. From the figure, we see that scores calculated from MUSHRA-like tests

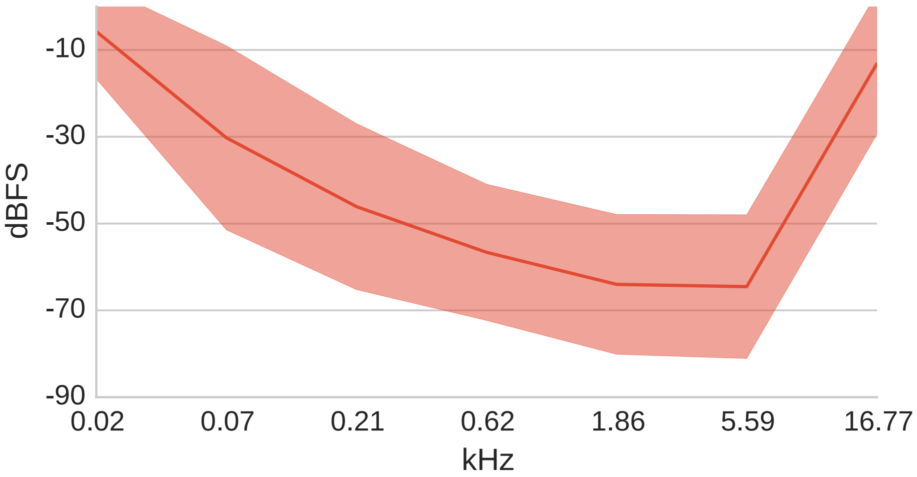


Figure 3.4. Mean in-situ hearing threshold of web-MUSHRA participants (N=520). Shaded region is +/- sample standard deviation.

on the web correlate well to lab-MUSHRA scores. For comparison, we also correlated the corresponding the BSS-Eval objective measures (SDR, ISR, SIR, and SAR [**42**])—all of which were less correlated to the lab-MUSHRA scores than the web-MUSHRA scores were.

Since Pearson correlation only measures if there is a linear relationship and has no sense of scale, we also calculated the Krippendorff's alpha coefficient between the web-MUSHRA scores and the lab-MUSHRA scores (see Figure 3.3b). Krippendorff's alpha is a scale-dependent statistic that measures the agreement among the participants' ratings and ranges from 0 (no agreement) to 1 (perfect agreement) [**96, 64**]. Since Krippendorff's alpha is a scale-dependent statistic, BSS-Eval measures were not compared since they are not on a comparable scale. Krippendorff's alpha is a more conservative measure than Pearson correlation, therefore the agreement values in Figure 3.3b are lower that in 3.3a, but the general trends are the same.

We also used the information from the hearing tests described in Section 3.4.1 to determine the participants whose abilities and environments are more similar to the 'ideal' lab-based environment and expert participants. Using this information, we tried weighting participants' ratings to make aggregate scores more similar to those estimated by lab-MUSHRA. We created three variants of web-MUSHRA that use this participant data in different ways.

The first variation, *Weighted Web-MUSHRA*, uses the data from the *in situ hearing response estimation* described in Section 3.4.1. 10 of the 530 participants' responses were rejected according to the criteria in Section 3.4.1. Figure 3.4 shows the mean and standard deviation of the responses that met the inclusion criteria. The graph is encouragingly resemblant of minimum audible sound level curves [**123**]. We combined a participant's resulting hearing-threshold curve with the power spectral densities of the stimuli; thereby creating a weight, *filtered-psd-rms*, that is higher when the stimulus contains audible frequency content and lower when it contains inaudible frequency content. For example if a participant is listening over low-quality headphones that have poor high-frequency response yet have adequate low- and mid-frequency response, the participant could still meaningfully assess the audio quality of speech recorded with an 8kHz sampling rate. To calculate this weight, we first subtracted the log hearing threshold (linearly interpolated to N (2048) frequency bins), $\mathbf{H}_k$, of the kth participant, from the log power spectral density, S_m of the mth mixture. This inverse filters the power spectral density, emphasizing the frequencies that the participant can hear well. We then take the log root mean square (RMS) of this difference to obtain a weight, $w_{m,k}$ (see 3.1). This weight is higher when the stimulus contains frequency content the participant can hear well, and lower when it contains frequency content that the participant cannot hear well. Using these weights, stimulus scores were calculated using a weighted median.

$$w_{m,k} = 20\log_{10}\sqrt{\frac{1}{N^2}\sum_{n}^{N} 10^{(\mathbf{S}_{m,n}-\mathbf{H}_{k,n})/10.0}} \tag{3.1}$$

The second variation, *Screened Web-MUSHRA*, is the same as web-MUSHRA but using only responses from participants who passed the *hearing screening* described in Section 3.4.1. The third variation, *Weighted and Screened Web-MUSHRA*, combines both approaches, screening participants and weighting them based on Equation 3.1 (again using the weighted median).

Figure 3.3 also shows the correlation and agreement of the scores estimated by the web-MUSHRA variations and the scores estimated by lab-MUSHRA. The hearing-screened variations increased correlation with lab-MUSHRA in all qualities except *suppression of other sources*, but this difference was only statistically different for *overall quality* (Bonferroni-adjusted $p < 0.05$ according to a William's t-test[2] [**197**]). Also, for *absence of artificial noises*, the hearing-screened variations were the only variations whose lab-MUSHRA correlations were statistically significantly different than BSS-Eval (William's t-test $p < 0.05$). However, the hearing-response weighting did not seem to affect the correlation. A possible explanation for these minimal and absent effects is that the audio source separation algorithms under test have such easily detectable impairments that they can be heard and assessed in both good and poor listening conditions—a positive quality for crowdsourced evaluations.

[2] The William's t-test is a statistical test that can be used for measuring the difference between overlapped, dependent correlations.

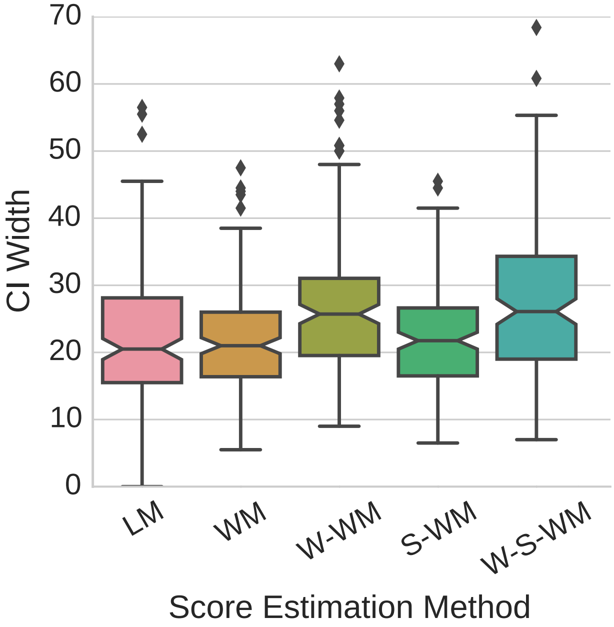


Figure 3.5. Box plots of the 95% confidence-interval widths of MUSHRA scores for systems under test (i.e. excluding reference and anchors) for each quality. Boxes are the $Q1$ to $Q3$ quartiles. Notches are 95% CI on $Q2$. Whiskers are $1.5 * (Q3 - Q1)$ extensions. **LM**=‘Lab-MUSHRA’, **WM**=‘Web-MUSHRA’, **W-WM**=‘Weighted Web-MUSHRA’, **S-WM**=‘Screened Web-MUSHRA’, **W-S-WM**=‘Weighted and Screened Web-MUSHRA’

3.4.3.3. Are lab-MUSHRA scores more discriminative than web-MUSHRA scores? Regardless of the correlation or agreement, it may be that scores calculated from MUSHRA-like tests on the web are not as discriminative and have wider confidence intervals than the gold-standard, lab-based MUSHRA tests. Tighter confidence intervals are preferable because of their greater statistical power in discriminating between stimulus scores. To investigate this, we calculated the widths of the 95% confidence intervals for the scores of the systems under tests for all four quality scales and pooled them together into one distribution for each test location (lab and web). Since there were 20 lab participants, we limited ourselves to using only the first 20 web participants for each sample group. The confidence-interval widths for the web-MUSHRA (without screening and weighting) and lab-MUSHRA scores are very similar—they have almost identical sample means (web: 22.0, lab: 21.9), and the sample standard deviations of the web score widths are just a bit smaller than the lab score widths (web: 7.4, lab: 10.3). A two-sided paired t-test fails to reject the null hypothesis that the means are equal ($t(159) = 0.35$, $p = 0.73$). In Figure 3.5, we plot the CI-width box plots of the MUSHRA variations by quality type. A one-way analysis of variance (ANOVA) on the test type (i.e. MUSHRA variation) rejected the null hypothesis that the means are equal ($F(4, 955) = 8.2$, $p < 0.001$). A post hoc Tukey honest significant difference (HSD) test on the MUSHRA variations showed that weighted variations had statistically significantly larger confidence intervals than the other groups at a $p < 0.05$ significance level. This analysis implies that estimated scores from MUSHRA-like tests on the web can have similarly sized confidence intervals as those from MUSHRA in the lab but that the weighting described in Section 3.4.3.2 may increase confidence intervals of scores.

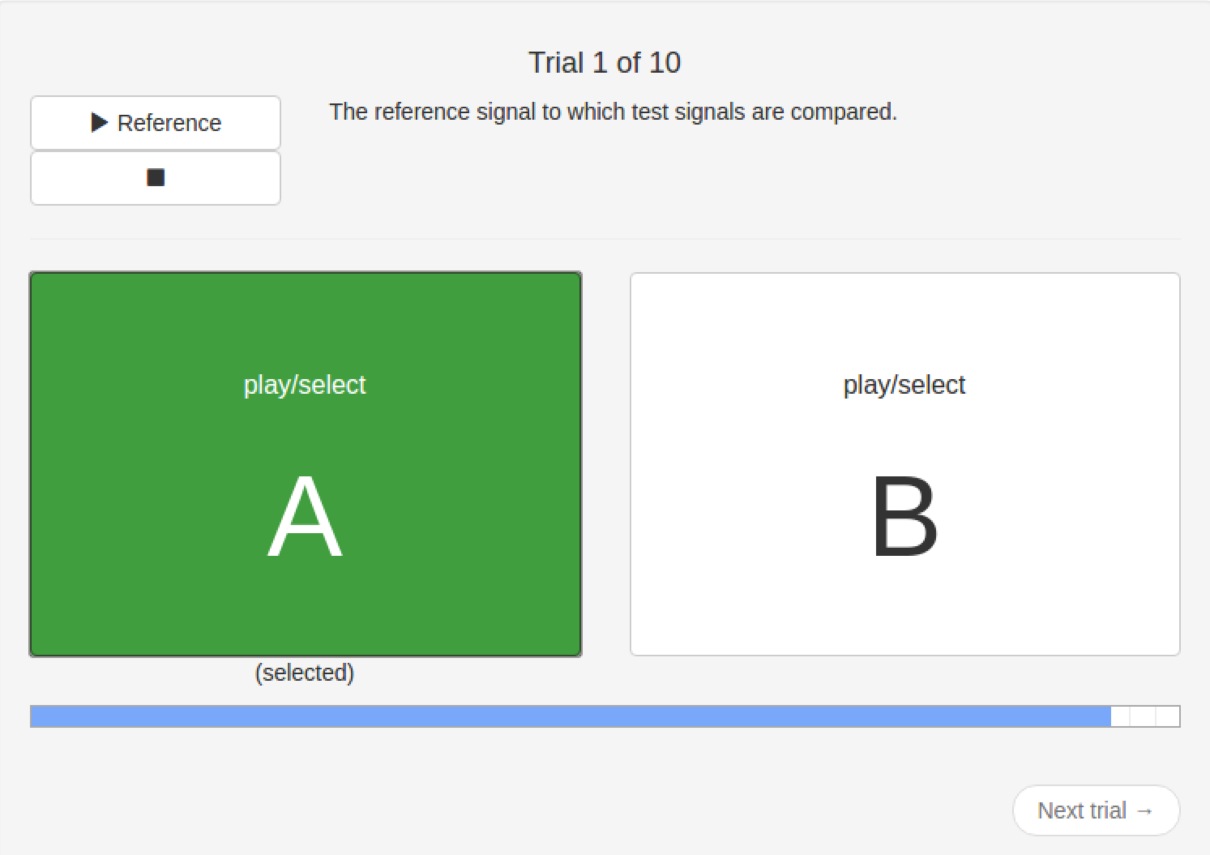
3.4.3.4. How much time and money is required to obtain web-MUSHRA results similar to "gold-standard" lab-MUSHRA listening tests? We were able to collect web-MUSHRA evaluations from 530 participants in only 8.2 hours and paid them a total of $1040 USD (plus Amazon's fee, which was 10% of the this total reward when we collected this data in June 2015). However, from our analysis it seems reasonable to use only the first 20 participants for each condition. This would lower the cost to $354 (plus Amazon's fee).

3.4.4. Discussion

Despite lacking control over listening environments, hearing abilities, and expertise, the scores from web-based MUSHRA-like evaluation were surprisingly similar to the lab-based MUSHRA evaluation. For example, the web-based and lab-based MUSHRA scores had a Pearson correlation of $r = 0.96$ for the well-defined quality scale *suppression of other sources* and $r = 0.69$ for the poorly-defined quality scale *absence of artificial noise.* The distributions of confidence-interval widths for the lab-MUSHRA and web-MUSHRA score estimates were also very similar, which tells us that web-MUSHRA quality estimates are not noisier or less discriminative than the estimates from the lab-MUSHRA test. Rather, we can instead interpret the web-MUSHRA scores as slightly different, "real-world" quality scores. Additionally, by using a hearing screening test, we can make the scores more similar to the lab-MUSHRA scores—e.g., this raised the correlation between web-based and lab-based MUSHRA scores from 0.78 to $r = 0.89$ for *overall quality.*

3.5. Crowdsourced Web-based Pairwise-Comparison Listening Tests

As described in Section 3.2, pairwise-comparison listening tests do not have the same limitations as MUSHRA listening tests and may be more suited to crowdsourced testing than MUSHRA. In this section, we first provide an overview of pairwise-comparison tests and describe the details of the experiment. We then describe the probabilistic models for estimating continuous scores from discrete comparisons, details of how we fit the models, and an analysis of the results.



Trial 1 of 10

The reference signal to which test signals are compared.

Figure 3.6. The interface for the pairwise-comparison listening test running on the CAQE software.

3.5.1. Overview of pairwise-comparison tests

In a typical pairwise-comparison test, a participant is given two test stimuli and possibly one or more references in each pairwise comparison and is asked to choose which of the two test stimuli are higher on a given quality scale, e.g. "In which recording are the drums louder? Recording A or B?". A participant may answer the same question for several different pairs of stimuli, and each pair will be evaluated by several different participants. Using these pairwise preference decisions, a stimulus preference order and possibly stimulus scores may be estimated. Tests of this form have a long history in psychometric testing and are well studied [**32**].

There are common, lab-based pairwise-comparison tests for audio, but they are not appropriate for our evaluation context. For instance, an ABX test is a pairwise test in which a participant is given three stimuli: a reference and two unknown stimuli. One of the unknown stimuli is the same as the reference stimulus, and the participant's task is to choose which of these unknown stimuli is the same as the reference. An ABX test can be used to statistically test if the output of the two audio systems or algorithms are perceptually different. Extending this methodology, Recommendation ITU-R BS.1116-2[**81**] is a scaled pairwise-comparison test that is used to evaluate audio with very small, barely-discernible impairments (small differences from the reference audio). ITU-R BS.1116-2 is similar to ABX, but instead of simply choosing which stimulus is the reference, the participant assesses the impairments on each stimulus compared to the reference on a continuous scale. However, this test is not appropriate when a reference is not available or when evaluating lower quality audio systems since it is "poor at discriminating between small differences in quality at the bottom of the scale" [**82**].

Fortunately, there are also pairwise-comparison models that can estimate meaningful rankings and scores of stimuli even when a reference is not available (Thurstonian [**186**] and Bradley-Terry [**14**]). These models estimate latent stimulus scores by fitting the model to the empirical probabilities of the pairwise-comparison outcomes (explained in more detail in Section 3.5.4.1). The probabilistic framework used in these models makes it easy to incorporate additional information (e.g., measures of reliability and hearing response estimates) into the score estimation. We will be focusing on one such model in our work.

3.5.2. Methods

To perform our pairwise-comparison listening test on the web, we again utilized Amazon's Mechanical Turk to recruit and pay participants as we did in web-MUSHRA. We also assigned HITs in the same manner as we did in the MUSHRA experiment—i.e. assigning participants to one of the four quality scales, and allowing each participant to perform up to 10 trials, one for each mixture (randomly ordered). However, for each HIT, instead of performing a MUSHRA trial, participants compared all pairs of stimuli associated with a mixture—$\binom{8}{2}$, or 28, pairwise comparisons. We limited each participant to one quality scale to reduce task confusion and to eliminate any quality scale ordering effects.

In each comparison, participants were shown the pairwise-comparison interface shown in Figure 3.6. Using this interface, when a participant first selects a stimulus (A, B, mixture, or reference), looped playback of the selected stimulus begins. When a participant selects a subsequent stimulus in a comparison, the playback loop synchronously switches to the selected stimulus, maintaining the current playback location. A participant can only proceed to the next comparison after they have listened to the stimuli for five seconds and have selected either the A or B stimulus. As in the web-MUSHRA training, participants were required to listen to examples of reference and anchor stimuli to familiarize the themselves with the quality scales. However, because participants were required to listen to each stimulus before making their selection, we did not require participants to listen to all of the condition's stimuli in the training before evaluating them (as is in MUSHRA).

All instructions were kept as similar to web-MUSHRA as possible while adapting them for the pairwise comparisons. Lastly, in their first trial, participants also completed a survey and two hearing tests as in the web-MUSHRA experiment (see Section 3.4.1). These hearing tests give us information about the participants' listening environments, hearing abilities, and reliability.

3.5.3. Data Collection and Overview

As in the web-MUSHRA data collection (see Section 3.4.2), we collected *a minimum of* 20 pairwise-comparison trials for each condition (mixture / quality scale pair) by participants that passed the hearing screening. For each condition, a participant compared all 28 pairs of stimuli. We paid participants \$0.80 for completing the first trial, which included the hearing tests, and \$0.50 for subsequent trials. In addition, participants could receive up to a \$0.25 bonus based on the consistency of their pairwise-comparisons. Only participants who had at least 1000 AMT assignments approved and a 97% approval rate were allowed to accept the HIT. Trial and participant statistics are shown in Table 3.3 along with the web-MUSHRA statistics for comparison.

According to the participant survey, the distribution of reported listening devices was 81% headphones, 10% laptop speakers, 8% loudspeakers. After the hearing screening this distribution changed to: 88% headphones, 3% laptop speakers, 8% loudspeakers.

81% of participants reported to never having participated in a listening-based study before. This confirms that this sampled AMT worker population is also primarily non-experts.

Table 3.3. Trial statistics of web-based listening tests: *web-MUSHRA* and *pairwise comparison*

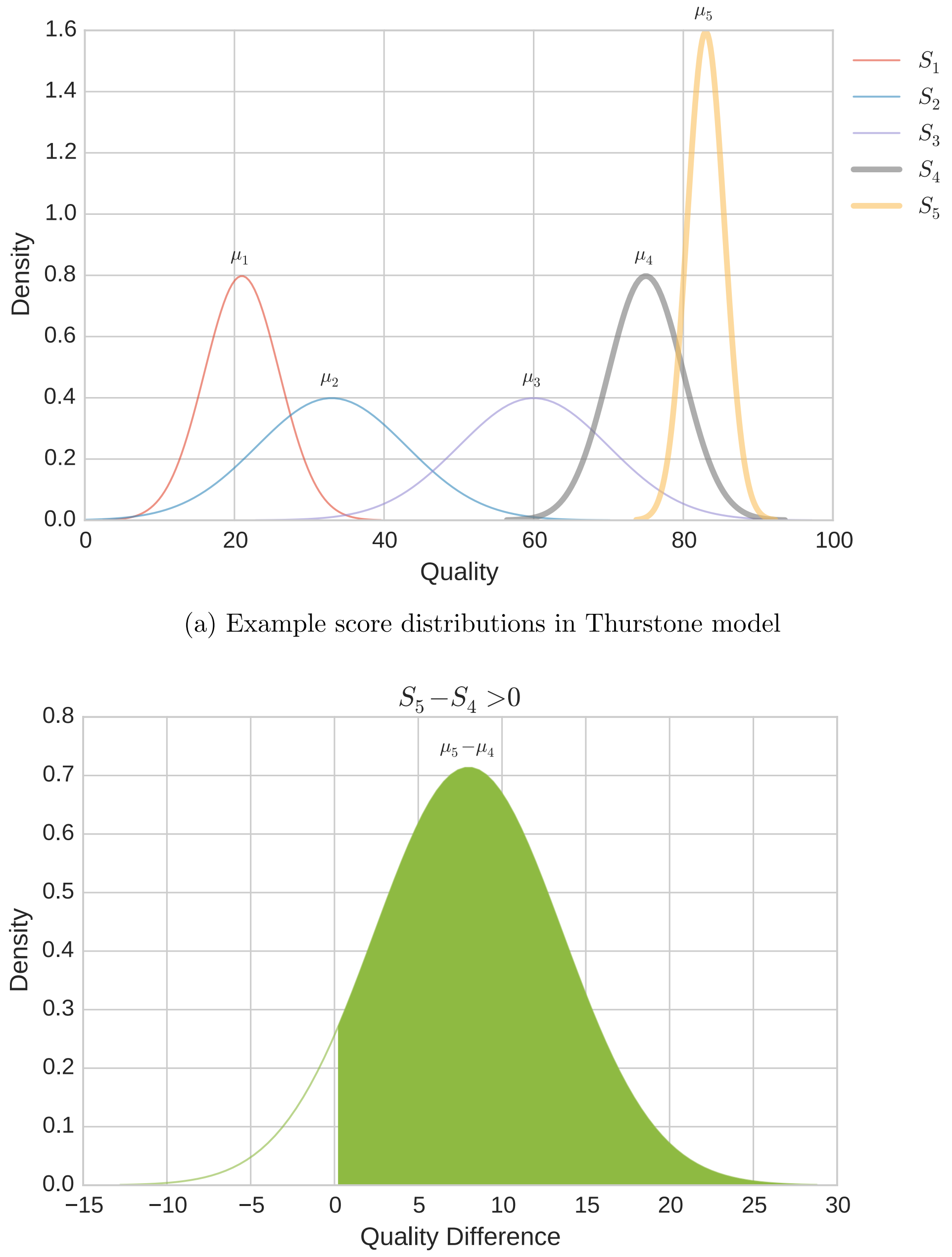
Listening test type	**web-MUSHRA**	**Pairwise**
No. of participants	530	458
No. of participants that passed hearing screening	336	345
Participant reported gender		
- No. female	184	173
- No. male	346	285
Participant reported age		
- min	18	18
- max	67	69
- mean	31	32.6
- median	28	30
No. of Trials	1763	1444
Trials per condition		
- min	26	25
- max	55	44
- mean	34.4	30.0
No. of trials by participants that passed hearing screening	1147	1113
Trials per condition by participants that passed hearing screening		
- min	20	20
- max	37	34
- mean	23.6	22.7
Trials per participant		
- min	1	1
- max	10	10
- mean	3.3	3.2

3.5.4. The Audio Evaluator Model and Related Models

In this section, we review the probabilistic models we will use to estimate the latent quality scores from the pairwise-comparison data. First we will review the Thurstone model, and then we will present our extension to the model: a probabilistic audio-evaluator model that puts more weight on "reliable participants" when estimating the latent scores. We extended the Thurstone model instead of the Bradley-Terry model because we would like to infer our score estimates' uncertainty, and the Thurstone model is better suited for Bayesian inference.

Our model takes the noisy data collected from the web-based participants and tries to combine it in such a way to reduce the noise and estimate scores that are more comparable to those estimated from lab-based participants' data. Our model draws inspiration from previous annotator models [**34, 175, 29, 136**] which model annotator reliability as either a latent parameter or estimate it from annotator performance on gold-standard data, and it directly builds on Chen et al's [**29**] extension to the Thurstone model.

3.5.4.1. Overview of Thurstone models. A Thurstonian model is a probabilistic latent variable model that maps discrete preference orderings of items (e.g., pairwise comparisons) to latent scores on an interval scale [**186**]. The model assumes that the items can be placed on an interval scale and there is some measurement error when people judge the scale value of a item. Therefore, we can treat the scale values of items as random variables. Figure 3.7a shows an example distribution of a set of items' $(a_1 \dots a_5)$ scale values $(S_n \dots S_5)$. The distance between two items on the unobserved scale and the measurement error affect the pairwise preference probabilities of the items—the larger the perceived difference between two items on the scale, the greater probability that the higher item on the scale will be preferred by a listener. Figure 3.7b illustrates this by plotting the density of the perceived difference between items a_4 and a_5. The model also makes the assumption that the measurement error around the true scale value for each item is Gaussian. In our case, the discrete preference orderings of items are the pairwise



(a) Example score distributions in Thurstone model

(b) Example difference in score distributions in Thurstone model

Figure 3.7. Example distributions of test items on an interval scale. Example distribution of the perceived difference between two test items, illustrating the probability that a participant prefers item a_4 over item a_5 from Figure 3.7a—i.e., $P(S_i > S_j)$ represented by the shaded positive region.

Table 3.4. Model notation definitions

$N > 0$	number of audio stimuli
$T > 0$	number of pairwise comparisons
$K > 0$	number of participants
a_n for $n \in 1 : N$	audio stimuli
h	hidden reference stimulus index
L	set of hidden anchor stimuli indices
$a_i \succ a_j$	event that audio stimulus a_i is chosen over audio stimulus a_j and therefore considered to be higher on the specified quality scale (e.g., "Overall Quality" or "Suppression of Other Sources")
μ_n	score (i.e., mean quality) of stimulus a_n
σ_n^2	variance of stimulus a_n quality
σ	global variance of stimulus quality
S_n	quality variable for stimulus a_n
$\eta_k \in [0 : 1]$	"reliability" of participant $k \in 1 : K$
Φ	normal cumulative distribution function
$ii[t], jj[t] \in 1 : N$	indices for stimulus pair in comparison $t \in 1 : T$
$kk[t] \in 1 : K$	index of the participant in comparison $t \in 1 : T$

comparisons of audio stimuli as described earlier, and our interval scale is an audio quality measure. The basic Thurstone model is as follows (see Table 3.4 for notation):

$$S_n \sim \text{Normal}(\mu_n, \sigma_n^2), \text{ for } n \in 1 : N \tag{3.2}$$

$$\Pr(a_i \succ a_j) = \Pr(S_i > S_j), \text{ for } i, j \in 1 : N \text{ where } i \neq j \tag{3.3}$$

$$= \Pr(S_i - S_j > 0) \tag{3.4}$$

As shown in [**189**], this can be rewritten as:

$$\Pr(a_i \succ a_j) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) \tag{3.5}$$

For simplicity and identifiability, it is common to assume that all σ_n^2 are equal. This variation is known as Thurstone Model V [**186**]. This simplifies Equation 3.5 to:

$$\Pr(a_i \succ a_j) = \Phi\left(\frac{\mu_i - \mu_j}{\sigma\sqrt{2}}\right) \tag{3.6}$$

3.5.4.2. Thurstone model for audio stimuli. Using Equation 3.6, we can relate a preference probability of two audio stimuli to their latent quality scores (μ_i and μ_j) and variance (σ). However, we need to estimate the scores of all audio stimuli. Using the model relationship in Equation 3.6, we can write the likelihood for a set of T pairwise comparisons:

$$\mathcal{L}(\theta | a_{ii[t]} \succ a_{jj[t]} \forall t \in 1:T) = \prod_{t=1}^{T} \Phi\left(\frac{\mu_{ii[t]} - \mu_{jj[t]}}{\sigma\sqrt{2}}\right)$$

$$\begin{aligned}
&\mu_n \sim \text{Uniform} \in [0, 100],\ \text{ for } n \neq h,\ n \notin L \\
&\mu_h \sim \text{Truncated-Normal}(100, 5) \in [0, 100] \\
&\mu_n \sim \text{Truncated-Normal}(15, 15) \in [0, 100],\ \text{ for } n \in L \\
&\sigma \sim \text{Uniform} \in [0, 100]
\end{aligned} \tag{\textbf{Thurstone model}}$$

where $\theta = (\mu, \sigma)$. Note that all of the preference information for a comparison is in the indices, $ii[t]$ and $jj[t]$, since the preferred stimulus is always assigned to $ii[t]$.

Since the MUSHRA scale range is $[0, 100]$, we set the priors for the stimuli of systems under test (all stimuli but the hidden reference and anchors) to Uniform$(0, 100)$ and limit the range of all other variables to $[0, 100]$. The priors on the hidden reference and anchor scores are set according to our expectations as well—very high for the hidden reference and relatively low for the hidden anchors.

We can then estimate the posterior probabilities of the scores, given a set of pairwise comparisons, using Bayesian inference with Markov chain Monte Carlo (MCMC) sampling [**73**].

3.5.4.3. Chen's extension. Chen et al [**29**] extended the basic Thurstone model for a crowdsourcing context in which potentially unreliable participants perform the comparisons. They modeled the probability that participant k would choose the higher quality stimulus, $\Pr(a_i \succ_k a_j | a_i \succ a_j)$, as a latent variable $\eta_k \in [0, 1]$. When a participant is entirely reliable, $\eta_k = 1$. When a participant chooses at random, $\eta_k = 0.5$, and when a participant is entirely "adversarial", $\eta_k = 0$. In this formulation, the participant reliability parameters are entirely latent. This allows us to incorporate data from all participants but put more weight on responses by participants that are in agreement with the "herd" (i.e., the other participants in agreement). The likelihood of this model is:

$$\mathcal{L}(\theta | a_{ii[t]} \succ a_{jj[t]} \forall t \in 1:T) = \prod_{t=1}^{T} \left[\eta_{kk[t]} \Phi\left(\frac{\mu_{ii[t]} - \mu_{jj[t]}}{\sigma\sqrt{2}} \right) + (1 - \eta_{kk[t]}) \Phi\left(\frac{\mu_{jj[t]} - \mu_{ii[t]}}{\sigma\sqrt{2}} \right) \right]$$

$$s.t. \ \frac{1}{K} \sum_{k} \eta_k > 0.5$$

$$\mu_n \sim \text{Uniform} \in [0,100], \text{ for } n \neq h,\ n \notin L$$

$$\mu_h \sim \text{Truncated-Normal}(100,5) \in [0,100]$$

$$\mu_n \sim \text{Truncated-Normal}(15,15) \in [0,100], \text{ for } n \in L$$

$$\sigma \sim \text{Uniform} \in [0,100]$$

$$\textbf{(Chen model)} \qquad \eta_k \sim \text{Beta}(\alpha_\eta, \beta_\eta) \in [0,1], \text{ for } k \in 1:K$$

where $\theta = (\mu, \sigma, \eta)$. We believe that participants on average are not adversarial, and we incorporate this belief in a negatively skewed Beta$(\alpha_\eta, \beta_\eta)$ prior (e.g., $\alpha_\eta > 1$, $\beta_\eta = 1$) on η and the constraint $\frac{1}{K}\sum_k \eta_k > 0.5$. Without this bias, the model would not be identifiable and could converge onto an equally probable but flipped solution (scores flipped around the scale midpoint of 50) in which the majority of users are adversarial. We describe our procedure for choosing α_η and β_η in Section 3.5.5.3.

Because the participant reliability variables are latent, a disadvantage of the Chen model is that it could possibly put more trust in a large subset of participants that may be in agreement but who are not truly reliable.

3.5.4.4. Our extension, the Audio-Evaluator Model. Our model addresses this disadvantage of the Chen model by using participant features to inform a prior on the participant reliability variables, η_k for $k \in 1:K$. We extend the Chen model by replacing the Beta$(\alpha_\eta, \beta_\eta)$ prior on each η_k with a Logit-Normal$(\mathbf{x}_k^\intercal \boldsymbol{\beta}, \gamma)$ distribution where $\mathbf{x}_k$ is an augmented vector of participant features for the kth participant and $\boldsymbol{\beta}$ are latent variable coefficients with Normal$(0,1)$ priors. γ is the variance of the η_k prior and allows us to balance how much η_k is influenced by $\mathbf{x}_k^\intercal \boldsymbol{\beta}$ and how much it is latent. All of the participant reliability variables are 2σ-standardized (i.e., mean centered and divided by two standard deviations). The likelihood of this model is:

$$\mathcal{L}(\theta | a_{ii[t]} \succ a_{jj[t]} \forall t \in 1:T) = \prod_{t=1}^{T} \left[\eta_{kk[t]} \Phi\left(\frac{\mu_{ii[t]} - \mu_{jj[t]}}{\sigma\sqrt{2}}\right) + (1 - \eta_{kk[t]}) \Phi\left(\frac{\mu_{jj[t]} - \mu_{ii[t]}}{\sigma\sqrt{2}}\right) \right]$$

$$\begin{aligned} \eta_{kk[t]} &= \frac{1}{1 + \exp\left(-\tau_{kk[t]}\right)} \\ \tau_{kk[t]} &\sim \mathcal{N}(\mathbf{x_{kk[t]}}^{\intercal}\boldsymbol{\beta}, \gamma) \\ s.t.\ &\frac{1}{K}\sum_{k} \eta_k > 0.5 \end{aligned}$$

$$\begin{aligned} \mu_n &\sim \text{Uniform} \in [0, 100],\ \text{for } n \neq h,\ n \notin L \\ \mu_h &\sim \text{Truncated-Normal}(100, 5) \in [0, 100] \\ \mu_n &\sim \text{Truncated-Normal}(15, 15) \in [0, 100],\ \text{for } n \in L \\ \sigma &\sim \text{Uniform} \in [0, 100] \\ \text{(\textbf{audio-evaluator model})} \qquad \boldsymbol{\beta} &\sim \mathcal{N}(0, 1) \end{aligned}$$

We use the following participant features ($\mathbf{x}_k$) to inform the model of a participant's reliability: *filtered-psd-rms*, *passed-hearing-screening*, and *transitivity-satisfaction-rate.*

The *filtered-psd-rms* feature is calculated using the data from the *in situ hearing response estimation* described in Section 3.4.1. 10 of the 458 participants' responses were rejected according to the criteria in Section 3.4.1. Figure 3.8 shows the mean and standard

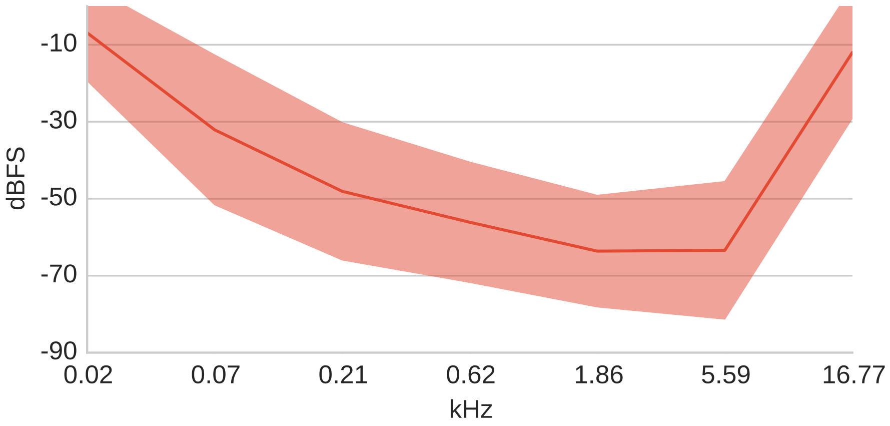


Figure 3.8. Mean in-situ hearing threshold of pairwise-comparison participants (N=448). Shaded region is +/- sample standard deviation.

deviation of the responses that met the inclusion criteria. The graph is also encouragingly resemblant of minimum audible sound level curves [**123**]. We again combined a participant's resulting hearing-threshold curve with the power spectral densities of the stimuli as we previously did for web-MUSHRA in Section 3.4.3.2 (see Equation 3.1); thereby creating a feature, *filtered-psd-rms*, for participant k that is higher when the reference stimulus contains audible frequency content and lower when it contains inaudible frequency content.

The *passed-hearing-screening* feature is a binary feature that indicates whether the participant passed the *hearing screening* also described in Section 3.4.1. This feature indicates that the participant can hear an adequate range of frequencies and is reliable since the test has an objective answer. We also used this feature in the "screened web-MUSHRA" method (see Section 3.4.3.2) to filter participants when aggregating participants' ratings into quality scores.

Algorithm 1 Calculate *transitivity-satisfaction-rate* (TSR)

```
$M$ is an $N$ by $N$ matrix where $M[i,j] == 1$ if $a_i \succ a_j$.
function CalculateTSR($M$)
    $n\_test \leftarrow 0$
    $n\_pass \leftarrow 0$
    for all $i, j, k \in 1 : N, i \neq j \neq k$ do
        if $M[i,j] == 1$ and $M[j,k] == 1$ then
            $n\_test \leftarrow n\_test + 1$
            if $M[i,k] == 1$ then
                $n\_pass \leftarrow n\_pass + 1$
            end if
        end if
    end for
    $tsr \leftarrow n\_pass / n\_test$
    return $tsr$
end function
```

The *transitivity-satisfaction-rate* feature, borrowed from [**28**], measures a participant's consistency by calculating the fraction of stimulus triples in which the participant violates transitivity in their pairwise comparisons (see Algorithm 1 for details). For example, if a participant chooses stimulus A over B, and B over C, then by transitivity we expect them to choose A over C as well—if they don't, then they have violated transitivity.

By using these participant features, we can set priors on the η_k variables, informing the model which participants are likely to be reliable and able to hear the stimuli. While our model could still suffer if the majority of the users were adversarial, this model puts more weight on users that we expect to be putting forth effort and that are in a good listening environment.

Table 3.5. Means and 95% Confidence Intervals of Pairwise Transitivity Statistics by Quality Scale (N=10)

Quality Scale	Average-TSR	WST
Overall Quality	0.91 [0.90, 0.93]	0.97 [0.90, 0.99]
Target Preservation	0.90 [0.88, 0.92]	0.97 [0.95, 0.99]
Suppression of Other Sources	0.92 [0.90, 0.93]	0.99 [0.96, 1.00]
Absence of Artificial Noises	0.91 [0.90, 0.92]	0.99 [0.97, 1.00]

3.5.5. Fitting the Models

3.5.5.1. Aggregate Transitivity of Responses. As previously mentioned, Thurstone models (see Section 3.5.4.1) model the perceived quality of a stimulus as a score on a unidimensional interval scale with added noise due to measurement error and variations between individuals. The estimated mean qualities (the scores) are inherently transitive (i.e., $(\mu_i > \mu_j) \wedge (\mu_j > \mu_k) \implies (\mu_i > \mu_k)$) because they are on a unidimensional interval scale.

If we expect a Thurstone model to fit the data well, then transitivity should also be present in the observed input data of the model. This observed input data is not just the preference data of an individual, but rather of a group of individuals which may not be in perfect agreement. To measure the aggregate transitivity of all participants, we calculate a "weak stochastic transitivity" measure (WST) from the empirical pairwise preference probabilities [**28, 149**]. This measure is similar to the *transitivity-satisfaction-rate* (TSR) that we calculated in Section 3.5.4.4, but it is a stochastic measure of the aggregate transitivity of the group rather than the deterministic transitivity of an individual participant. We calculate this measures as follows:

Let $\hat{P}(a_i \succ a_j)$ be the empirical probability that audio stimuli a_i was chosen over a_j. Then if $\hat{P}(a_i \succ a_j) \geq 0.5$ and $\hat{P}(a_j \succ a_k) \geq 0.5$ then *weak stochastic transitivity (WST)* is satisfied if $\hat{P}(a_i \succ a_k) \geq 0.5$.

This satisfaction criteria is then evaluated for all stimulus pairs in a test condition as we did in Algorithm 1 for each participant's transitivity satisfaction rate (TSR), and the fraction of satisfaction is the WST.

Table 3.5 shows each quality's mean WST over the 10 mixtures as well as the average participant transitivity satisfaction of responses (Average-TSR). If WST is violated, it's often an indication that the stimulus quality is actually multidimensional and participants have different strategies for dealing with the multiple dimensions, resulting in conflicting ratings. Except for a few violations, WST is satisfied for our pairwise-comparison data with *Overall Quality* having the lowest mean, which is as expected since this measure is likely multidimensional in nature.

3.5.5.2. MCMC Sampling. To estimate stimulus scores from our pairwise-comparison data, we fit all three models described in Section 3.5.4: the original Thurstone model (*Thurstone*), Chen's extension to the Thurstone model (*Chen*), and our extension to the Thurstone model: the Audio-Evaluator model (*Audio-Evaluator*). All models were fit to the pairwise-comparison data using the PyStan package [**37**] and the No-U-Turns Sampler [**73**] (NUTS) algorithm for MCMC sampling [**73**]. When fitting models, we drew two chains of 10,000 samples from the posterior distribution, dropping the first 5,000 and thinning by a factor of 2. Gelman and Rubin's potential scale reduction $\hat{R}$ is an MCMC sampling convergence diagnostic based on the within-chain and between-chain variance of two or more sampling chains [**54**]. It is generally accepted that chains are adequately

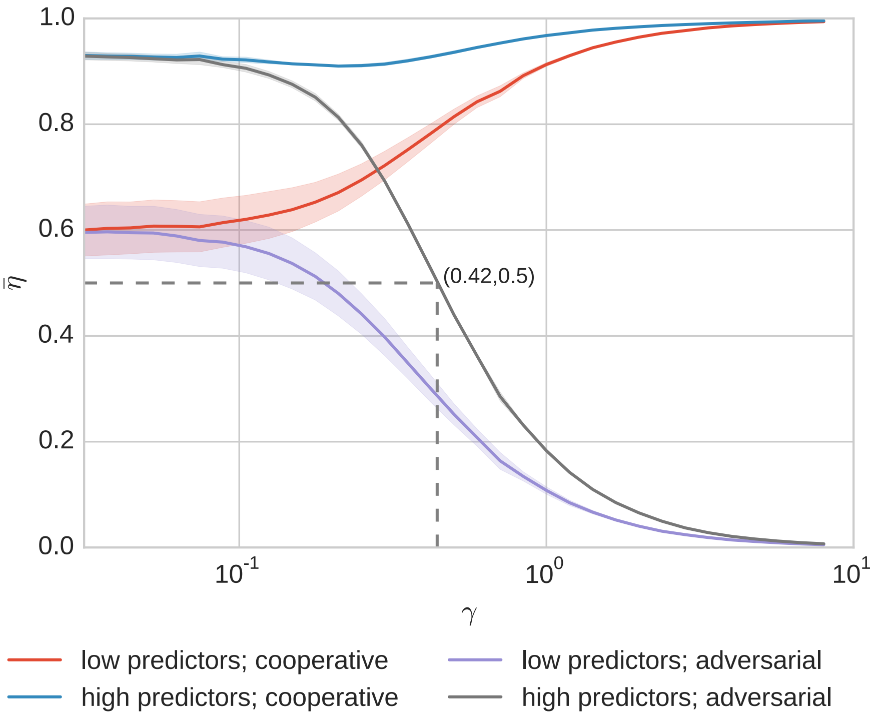


Figure 3.9. Four artificial participants' posterior η means. We fit the Audio-Evaluator model with the data of four artificial participants and 50 actual participants while varying γ.

mixed and sampling has converged when $\hat{R}$ is near 1.0 with an acceptance threshold of 1.1 [**54**]. After sampling, the variables for all models and conditions met the $\hat{R} < 1.1$ acceptance criteria.

3.5.5.3. Choosing η's scale parameters. Both the Chen model and the Audio-Evaluator model utilize priors on η, the participant reliability, that can be tuned based on our belief of the participant population's reliability. We in general expect participants to be more reliable and cooperative than adversarial, and therefore we utilize negatively skewed priors. For the Chen model, η is entirely latent and the prior on η is a negatively skewed Beta distribution. For the Audio-Evaluator model, we utilize the features described in

Section 3.5.4.4 to inform the prior on η, but the strength of this prior can be tuned using the model's γ parameter (the variance of the Logit-Normal distribution).

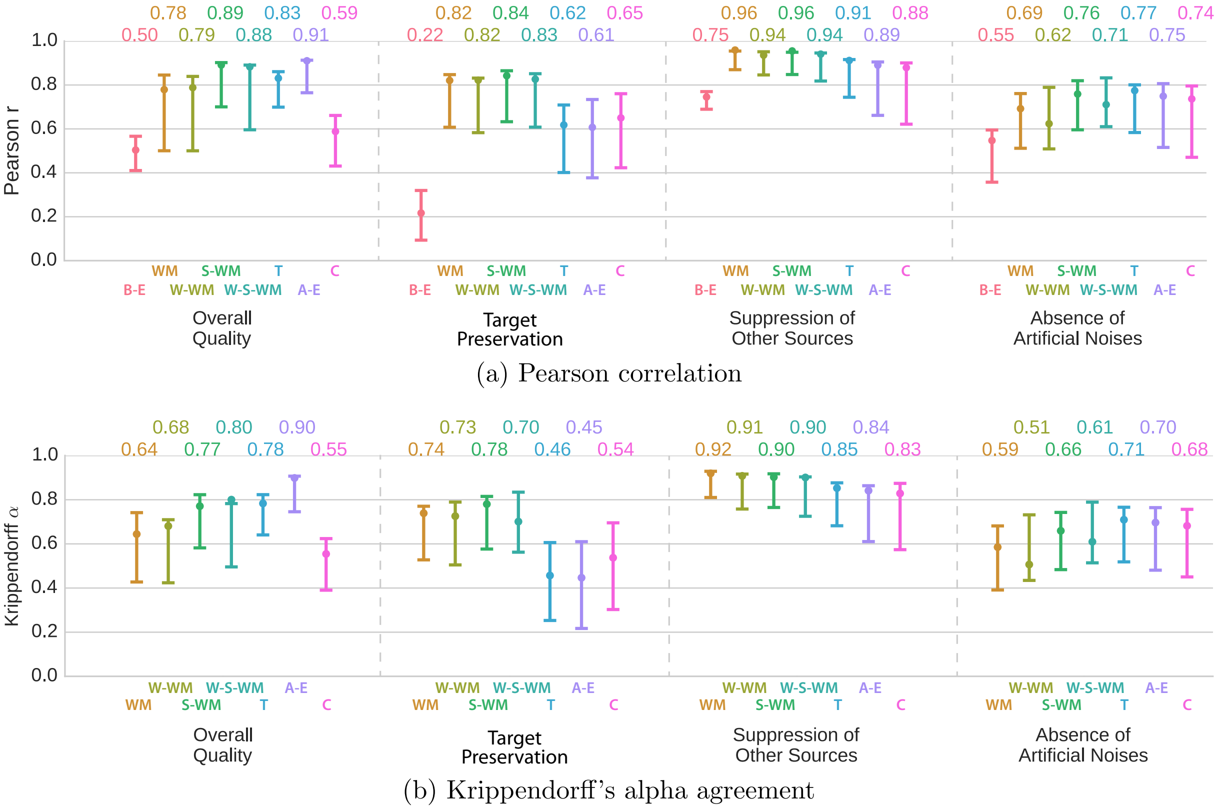
To choose a value for γ, we first fit the Audio-Evaluator model several times while varying γ between 2^{-5} and 2^3. We fit the model using pairwise comparisons with the data from 20 of the AMT participants. In addition, we added in synthetic data for four artificial participants which varied in high/low reliability features (i.e., predicted reliability) and cooperative/adversarial responses. We augmented the real AMT participant data with this synthetic data in order to observe how different values of γ would affect participants' estimated reliability (η) when participants have various reliability indicators in their data. To generate high reliability features, we used the maximum-valued *filtered-psd-rms*, *transitivity-satisfaction-rate*, and *passed-hearing-screening* features of the 20 participants, and for the low reliability features, we used the minimum feature values of the participants. To generate cooperative responses, we chose the stimulus with highest empirical preference probability for each pair, and to generate adversarial responses we chose the stimulus with the lowest empirical preference probability for each pair. Figure 3.9 displays the mean posterior η for each of these four artificial participants in response to varying γ—we can see how small values of γ put a lot of weight on the value predicted by the reliability features, and large values of γ put less weight on the value predicted by the reliability features, making the η in the Audio-Evaluator model behave more like the completely latent η in the Chen model.

We decided that a reasonable choice for γ is 0.42—a value at which the mean of η is 0.5 for an adversarial participant with high reliability features. At this value, η is flexible enough to ignore an adversarial participant, while still differentially-weighting cooperative users by reliability features. With this γ value for the Audio-Evaluator model, we then chose appropriate values for α_η and β_η in the Chen model to obtain a comparable overall prior on reliability. To do this, we drew 10,000 samples from the Audio-Evaluator model's η prior distribution, and then we computed the maximum likelihood estimates for α_η and β_η that fit a Beta(α_η, β_η) distribution to the samples. We estimated $\hat{\alpha}_\eta = 2.45$ and $\hat{\beta}_\eta = 1.35$—values that are also quite similar to those in the distribution used in the original Chen paper, Beta(2, 1). We used our estimated γ, α_η, and β_η values for the remainder of the paper to fit the models.

3.5.6. Results

3.5.6.1. Did the participants understand the task and rate reasonably? Since the data from participants is binary pairwise preference data, we cannot simply check the ratings distributions of the five qualities' reference and anchor stimuli to ascertain whether participants understood the task as we did for web-MUSHRA in Section 3.4.3. Instead, we checked that the empirical preference probability of reference stimuli were at least as high as other non-anchors, and that the empirical preference probability of anchor stimuli were lower than non-anchor stimuli. All qualities passed this check.

3.5.6.2. How similar are the three pairwise-comparison models' scores to lab-MUSHRA scores? To answer this question and to establish if our AMT-based pairwise-comparison listening test can act as a proxy to a lab-based, gold-standard test, we calculated the Pearson correlation between the lab-MUSHRA scores and the scores estimated from the pairwise-comparison tests and models for the systems under test. The results are shown in Figure 3.10a. For comparison, we also included the scores estimated by



(a) Pearson correlation

(b) Krippendorff's alpha agreement

Figure 3.10. Pearson correlation and Krippendorff's alpha agreement with the lab-MUSHRA scores and the scores estimated from the pairwise-comparison tests and models (web-MUSHRA and BSS-Eval included for comparison) for the four quality scales. Scores were limited to the systems under test (i.e., excluding the reference and anchors) and estimated using a sample size of 20 participants per mixture. Scores for all mixtures were concatenated before calculating the correlation and agreement for each quality scale ($N = 40$). Bars represent 95% CIs calculated from 1000 bootstrap iterations, randomly sampling with replacement from the lab-MUSHRA ratings and sampling from posterior distribution of the pairwise model scores. **B-E**='BSS-Eval', **WM**='Web-MUSHRA', **W-WM**='Weighted Web-MUSHRA', **S-WM**='Screened Web-MUSHRA', **W-S-WM**='Weighted and Screened Web-MUSHRA', **T**='Thurstone', **C**='Chen', **A-E**='Audio-Evaluator'

the web-MUSHRA variants and the scores calculated from the corresponding BSS-Eval measurements (i.e., SDR, ISR, SIR, and SAR). From the figure, we can see that the

Chen model has a lower correlation with lab-MUSHRA for *overall quality*—low enough that the we did not reject the null hypothesis that $r_{\text{BSS-Eval}} = r_{\text{Chen}}$ for that quality (William's t-test $p = 0.52$). In contrast, the *overall quality* scores estimated by the Audio-Evaluator model have the highest correlation with lab-MUSHRA ($r_{\text{Audio-Evaluator}} = 0.91$)—significantly higher than the *unscreened* variants of web-MUSHRA (William's t-test Bonferroni-adjusted $p < 0.05$).

When examining qualities other than *overall quality* however, we did not reject the null hypothesis that $r_{\text{Thurstone}} = r_{\text{Audio-Evaluator}} = r_{\text{Chen}}$ (William's t-test Bonferroni-adjusted $p > 0.05$ for all pairs of models for *target preservation*, *suppression of other sources*, and *absence of artificial noises*). When compared to the web-MUSHRA tests, the pairwise models had comparable lab-MUSHRA correlations for all qualities except *target preservation*, for which the correlations were lower.

As we did in the analysis of web-MUSHRA, we also calculated the Krippendorff's alpha coefficient between the scores estimated from the pairwise-comparison and the lab-MUSHRA scores (see Figure 3.10b) since Pearson correlation only measures if there is a linear relationship and has no sense of scale. Since Krippendorff's alpha is a scale-dependent statistic, BSS-Eval measures were not compared since they are not on a comparable scale. Krippendorff's alpha is a more conservative measure than Pearson correlation, therefore the agreement values in Figure 3.10b are lower that in 3.10a, but again, the general trends are the same.

3.5.6.3. Are lab-MUSHRA scores more discriminative than pairwise comparison scores? Regardless of the correlation or agreement, it may be that scores calculated

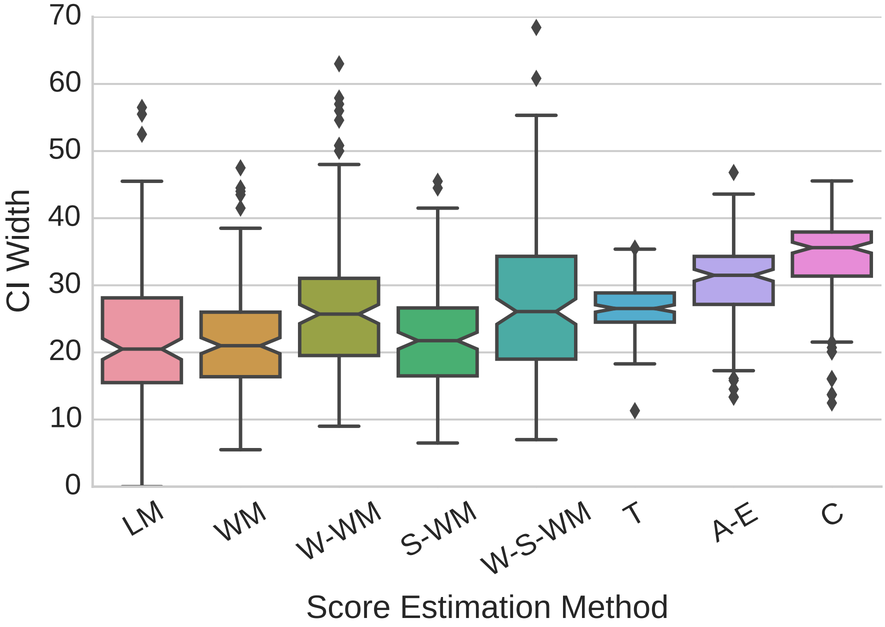


Figure 3.11. Box plots of the 95% confidence-interval widths of the web-based estimation scores and the lab-MUSHRA scores of a qualities ($N = 160$). The web-MUSHRA were variants included for comparison. Confidence-interval widths were limited to the systems under test (i.e., excluding the reference and anchors). Boxes are the $Q1$ to $Q3$ quartiles. Notches are 95% CI on $Q2$. Whiskers are $1.5 * (Q3 - Q1)$ extensions. **LM**='Lab-MUSHRA', **WM**='Web-MUSHRA', **W-WM**='Weighted Web-MUSHRA', **S-WM**='Screened Web-MUSHRA', **W-S-WM**='Weighted and Screened Web-MUSHRA', **T**='Thurstone', **C**='Chen', **A-E**='Audio-Evaluator'

from web-based pairwise-comparison tests are noisier and have wider confidence intervals than the gold-standard, lab-based MUSHRA tests. Tighter confidence intervals are preferable because of their greater statistical power in discriminating between stimulus scores. To investigate this, we calculated the widths of the 95% confidence intervals for the scores of the systems under tests for all four quality scales and pooled them according to their score estimation method. In Figure 3.11, we plot the CI-width box plots of the pairwise model scores by quality type. We again included the web-MUSHRA variants for comparison. Since there were 20 lab participants, we limited ourselves to using only the first 20 web participants for each sample group.

The pairwise-comparison method with the smallest CI-widths was the standard Thurstone model. The mean and standard deviation of the CI-widths of that model were $mean = 26.7$, $SD = 3.6$. The mean and standard deviation of the CI-widths of the lab-MUSHRA test were $mean = 22.0$, $SD = 10.3$. In Section 3.4.3.3, a t-test did not reject the null hypothesis that the means of the web-MUSHRA and lab-MUSHRA CI-widths were equal. However, a one-way ANOVA of the CI-widths by score estimation method (e.g., web-MUSHRA, lab-MUSHRA, Thurstone, etc.) rejected the null hypothesis that the means are equal ($F(7, 1272) = 46.0$, $p < 0.001$). A post hoc Tukey HSD test ($\alpha = 0.05$) on the score estimation method showed that each pairwise-comparison method had a statistically different CI-width mean than the other two pairwise-comparison methods as well as lab-MUSHRA, web-MUSHRA, and the screened web-MUSHRA. Therefore, when using an equal number of participants, web-MUSHRA was the most discriminative web-based listening test that we evaluated when tests were measured using confidence-interval width. In fact, web-MUSHRA's discriminative power is indistinguishable from lab-MUSHRA's. However, the pairwise-comparison tests were less discriminative than web-MUSHRA when using the same number of participants. Of the pairwise-comparison models, the Thurstone model was the most discriminative followed by the Audio-Evaluator and Chen models.

3.5.6.4. How much time and money is required to obtain pairwise-comparison results similar to "gold-standard" lab-MUSHRA listening tests? The cost to collect 1444 trials was \$1184 (plus Amazon's fee, which was 10% of the this total reward when we collected this data in June 2015). It took 35.5 hours to collect all of the trials for the pairwise data collection. This is longer than the web-MUSHRA data collection time (8.2 hours). Due to several participants complaining that the task was too long, we suspect the slower overall completion time resulted from an imbalance between task length and task reward. According to [**131**], AMT recruitment times (and therefore overall completion times) are dependent on reward. Higher payments yield faster overall completion times. We also collected more data than necessary and limited participants to conditions of only one quality scale. We were able to recruit participants at an average rate of 13 participants per hour, and this rate is actually much higher when a HIT is initially posted and then drops over time. Therefore recruitment isn't an issue. We could potentially reduce the completion time by up to a factor of four (down to 8.9 hours) by simply allowing participants to complete trials of conditions of more than one quality scale. Also, if the reward per task was a bit higher to appropriately match the length of the task and if only 20 trials per condition were collected, this completion time could be reduced even further. With a few small changes, the completion time of this listening test could likely be reduced to where a researcher could feasibly post a listening test before going to bed and wake up with results.

3.5.7. Increasing the Number of Participants per Condition in Pairwise Comparison Listening Tests

In Section 3.5.6, we analyzed the results of the pairwise-comparison listening tests given the data from just 20 participants. Are 20 participants enough? How many participants are required to maximize the aggregate consistency? How many participants are required

Table 3.6. Pearson Correlation to Lab-MUSHRA as Number of Participants Per Condition Increases

Model	Mean r at 20 participants	Mean max r	Mean no. of participants at max
Thurstone	0.91	0.93	33
Audio-Evaluator	0.90	0.93	38
Chen	0.89	0.94	60

to decrease confidence intervals enough to be comparable to the MUSHRA tests? Can we increase the correlation with the lab-MUSHRA scores if we increase the number of evaluations per stimulus pair?

To answer these questions, we collected more data for two of the mixtures (mixture 2, a speech mixture; and mixture 5, a music mixture) for all qualities, increasing the number of participants to 100 per condition (mixture / quality scale pair). In the following analysis, we revisit the analysis measures we used in Section 3.5.6 and observe how they change as we increase the number of participants. The ordering of the participants is the order that they each completed the trials. Note that because we only collected data for two of the mixtures, the values for 20 participants may be slightly different than the aggregate values presented earlier in Section 3.5.6 for all 10 mixtures.

3.5.7.1. Transitivity statistics. We measured the WST (defined in Section 3.5.5.1) of ratings as the number of participants increases (see Figure 3.12). As expected, WST of the ratings increases with more participants. The rates at which it increases levels off at around 50–60 participants—the increases in WST seem minimal after this threshold.

3.5.7.2. Comparing to the lab-based listening test results. As we increased the participants, we calculated the Pearson correlation between the lab-MUSHRA scores and the scores estimated from the pairwise-comparison tests and models for the systems under

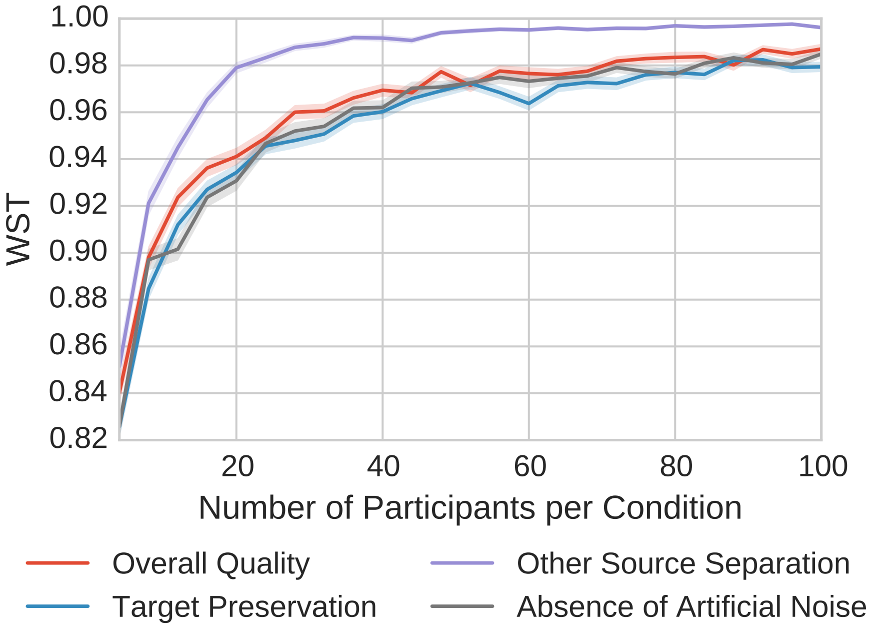


Figure 3.12. Mean WST stochastic transitivity measure over conditions as the number of participants increase. The bands are the 95% confidence intervals around the mean.

test. We estimated the scores at multiples of five participants per condition. We used the same process as described earlier in Section 3.5.6.2, but since we only have data for 2 mixtures, our correlations only have 8 data points rather than 40. Table 3.6 shows the mean correlation (over all four quality scales) at 20 participants per condition. It also shows the mean maximum correlation and the mean number of participants per condition at which the maxima occurred. The correlation leveled off quickly as the number of participants increased, and the correlation gain was minimal. This is asymptotic behavior was also observed in the estimated scores as we increased participants—there was typically a warm-up period that lasted for about 30–50 participants, after which the scores changed only subtly.

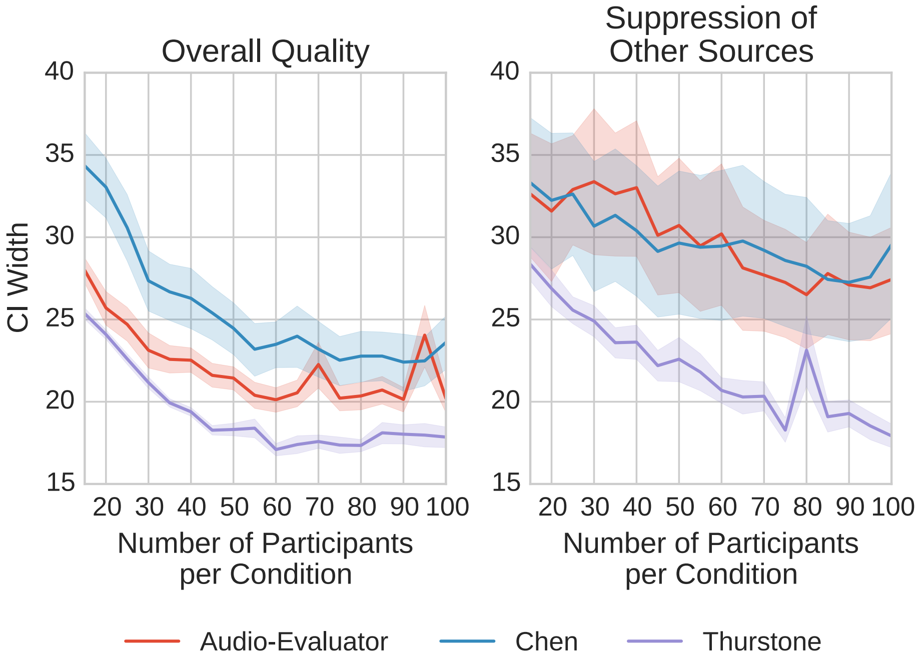


Figure 3.13. Mean confidence-interval width as number of participants per condition increases. The bands represent the 95% CI. N=8 for each mean.

3.5.7.3. Confidence intervals of scores. Lastly, as we increased the participants, we also calculated the CI widths for the scores of the systems under test. We used the same process as described earlier in Section 3.5.6.3. As would be expected, the confidence intervals get smaller as we increase the number of participants. The Thurstone model's CI-widths were the lowest for all qualities and its the trajectories for the qualities not shown lie between the trajectories for *overall quality* and *suppression of other sources*. While the Chen and Audio-Evaluator models' CI-widths are about the same for *suppression of other sources*, the Audio-Evaluator model's CI-widths are smaller than those of the Chen model for all other qualities (including those not shown).

To achieve a mean CI-width over all qualities equal to that of the 20-participant MUSHRA listening tests (22), the pairwise-comparison models would need evaluations from between 30–35 participants for the Thurstone model, 75–80 for the Audio-Evaluator model, and more than 100 for the Chen model.

3.5.8. Discussion

In this section, we presented and evaluated three pairwise-comparison models for audio quality evaluation. All of these models were variations of the Thurstone model which estimates continuous-valued scores from discrete pairwise comparisons. To make the pairwise-estimated scores comparable to MUSHRA-estimated scores, we informed all of the models of the score scaling by using priors on the reference and anchor stimuli scores. For the *overall quality* scale, the Chen model had the lowest correlation to the lab-MUSHRA estimated scores, and the Audio-Evaluator model had the highest. However, for all other quality scales, the models estimated scores with similar correlations to the lab-MUSHRA scores. The correlations of the Thurstone and Audio-Evaluator models were also comparable to those of web-MUSHRA except for the *target preservation* quality, for which they were lower. Unfortunately, the confidence intervals of the scores estimated by the pairwise models were on average larger than those estimated by MUSHRA listening tests for the same number of participants—the order from smallest to largest CIs was: Thurstone, Audio-Evaluator, Chen.

We also investigated how the number of participants per condition affects pairwise-comparison listening tests. While increasing the number of participants, we observed there was typically a 30–50 participant warm-up period that affected the aggregate transitivity measure, the estimated scores, and their correlation with lab-MUSHRA scores. The confidence intervals around the estimated scores decreased as the number of participants increased, and with around 30–35 participants, the mean CI-width for the Thurstone model was comparable to that of the MUSHRA listening tests. The confidence interval requirements and therefore the discriminatory power of the listening test are dependent on the tester's goals and the stimuli—systems that perform similarly will produce similar scores and therefore require more data points (e.g., participants) to show that one score is statistically different from another.

Without further testing, we can only speculate as to why the Audio-Evaluator model's scores were more correlated with lab-MUSHRA scores for *overall quality* but not the other qualities. It may be that participants' hearing ability and reliability are more important when assessing a multi-dimensional scale such as overall quality. In addition, it is unclear as to why the models estimated *target preservation* scores were less correlated with the lab-MUSHRA scores than web-MUSHRA. As shown in [**42**], some listeners (including some lab participants) misinterpret the *target preservation* and *absence of artificial noises* scales by not distinguishing between additive and subtractive distortions. It may be that these scale misinterpretations result in ratings discrepancies that Thurstonian models are not robust to, unlike the simple median used to aggregate the continuous ratings in MUSHRA. Such scale misinterpretations could easily be resolved by using a quality scale that is inclusive of all distortions (e.g., "lack of distortions to target").

In general though, the results for the pairwise-comparison listening tests are very encouraging. These results establish that crowdsourced pairwise-comparison tests can also produce results similar to lab-based MUSHRA and can therefore be used when MUSHRA tests are not appropriate (e.g., when there isn't a reference, or there are more than 12 stimuli, etc.).

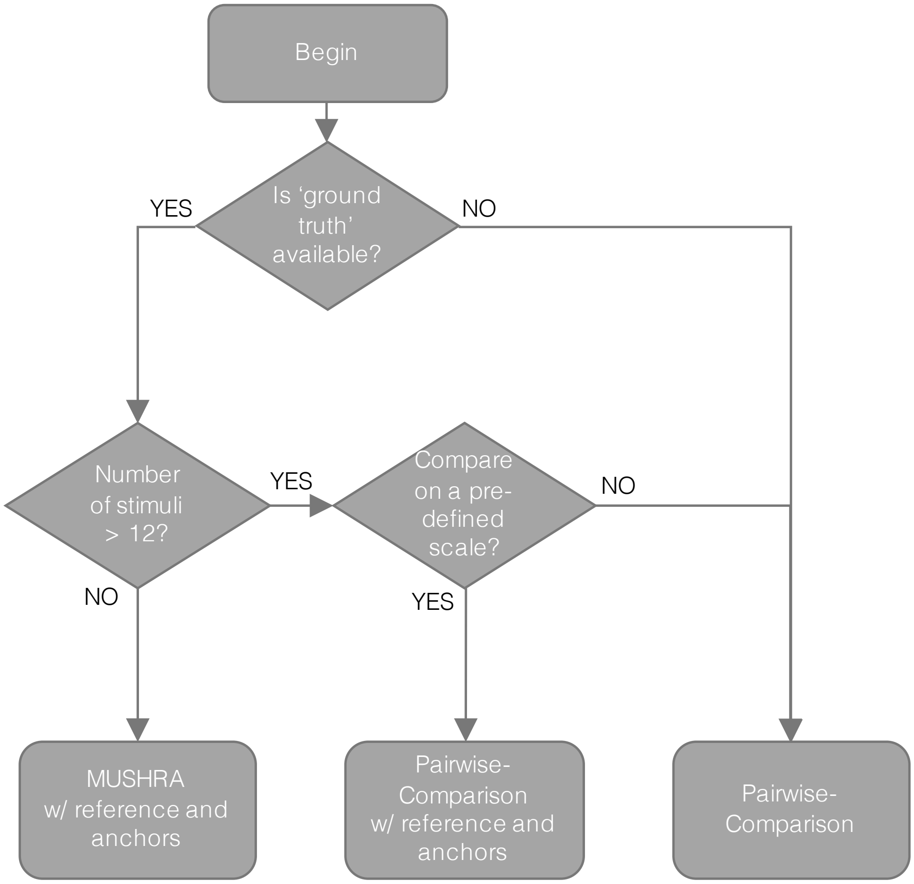

Begin
Is 'ground truth' available?
YES
NO
Number of stimuli > 12?
YES
Compare on a pre-defined scale?
NO
NO
YES
MUSHRA w/ reference and anchors
Pairwise-Comparison w/ reference and anchors
Pairwise-Comparison

Figure 3.14. Decision tree for deciding what type of listening test to use.

3.6. Recommendations

The choice of listening test depends on the needs of the researcher. Figure 3.14 presents a simple decision tree for deciding which listening test to use. The primary questions one needs to ask are:

1. Are there more than 12 stimuli (including anchors and references)? Comparing more than 12 stimuli (including the hidden reference and anchors) is beyond the limits of MUSHRA tests, but not beyond the limits of pairwise-comparisons tests.

Therefore, if there are more than 12 stimuli to evaluate, the stimuli should be evaluated using a pairwise-comparison listening test.

2. Are ground-truth audio signals available (e.g., original sources in the audio source separation task)? MUSHRA requires reference and anchor stimuli, which are generated from ground-truth data for most audio tasks. These stimuli essentially define the scale on which to evaluate other stimuli; therefore without reference and anchor stimuli, participants' ratings may be scaled differently and cannot reliably be aggregated. Without ground-truth audio, they cannot perform a MUSHRA listening test. If ground truth is not available, researchers should evaluate the stimuli using a pairwise-comparison listening test to obtain discriminative score estimates.

3. Is it necessary to compare scores on a pre-defined scale? If not and the investigators simply need a ranked ordering or want to evaluate several stimuli on an newly-defined scale in comparison to each other, a pairwise-comparison test can accommodate their needs without the requirement for ground truth. However, if it is necessary to compare to scores on a pre-defined scale (e.g., to previous MUSHRA tests), then the stimuli can be evaluated either with a pairwise-comparison or a MUSHRA listening test, but when evaluating with either these tests, a hidden reference and anchors must be included in the stimuli in order to scale the ratings.

Unless the answers to the above questions recommend a pairwise-comparison test, web-MUSHRA should be used since the estimated scores have tighter confidence intervals when using fewer participants.

In addition, if the stimulus assessment scale is an overall quality scale, our results demonstrate it is beneficial to use a hearing screening to filter participants for web-MUSHRA testing or to incorporate reliability data into the model for pairwise-comparison testing (i.e., the Audio-Evaluator model). Otherwise, we recommend using the standard Thurstone model if a pairwise-comparison test is necessary. It could be that the Audio-Evaluator model would also aid in evaluating stimuli with very high or low frequencies or high similarity which requires excellent listening conditions to detect the small changes. However, more testing would be necessary to determine this since the stimulus differences in this paper's experiments were easily detectable.

In regard to how many participants to whom the test should be administered, we recommend 20–30 for web-MUSHRA and 35–50 for pairwise-comparison tests for stimuli such as in the PEASS data set. However, the actual number will depend on both the number of stimuli one is testing and the confidence-interval requirements of the scores given the stimuli.

3.7. The CAQE Toolkit

The CAQE Toolkit is a set of software tools that implements the audio evaluation framework presented in this paper (see Figures 3.1 and 3.6). The CAQE Toolkit can be configured to run either web-MUSHRA or pairwise-comparison tests with or without the additional hearing tests presented in Section 3.5.2. It is written as a web application with a Flask/Python back-end and a HTML5/JavaScript front-end. It includes documentation and tools for easily deploying via Heroku (to eliminate the need for server-admin knowledge), managing workers and tasks from Amazon's Mechanical Turk, and analyzing the results. The CAQE Toolkit can be downloaded at `http://github.com/interactiveaudiolab/CAQE`.

3.8. Teaching Equalization Concepts with CAQE and SocialEQ

In this section, I'm going to briefly explain how CAQE could be used in conjunction with SocialEQ to overcome some of SocialEQs limitations. As stated earlier in Section 3.2.1, one of the problems with SocialEQ is that we may be learning meaningless correlations with frequency bands that the user cannot even hear (depending on their listening environment). With 40 audio examples (25 unique, 15 repeated), it is also difficult for listeners to rate consistently on a newly defined *continuous* scale since it requires rating in relation to all previously rated examples. Lastly, when we calculate the mean of the probability distribution in descriptor definition when mapping to the parameter space, we may end up with a poor aggregate relative spectral curve if not all contributors are in agreement and the distribution is multimodal. CAQE can overcome these limitations.

To learn an equalization concept with SocialEQ and CAQE, we could do the following. First we would break up the pairwise comparisons to reduce the amount of time any one individual spends rating. Since there are 25 unique audio examples in SocialEQ, we would break up the $\binom{25}{2}$, 300, pairwise comparisons across 4 individuals—75 per individual. Because listeners are just listening to overall timbre changes rather than listening for artifacts or sound suppression, it is faster to evaluate equalization examples than evaluating source separation. We expect 75 comparisons to take between 5 and 10 minutes. In the pairwise comparison task, we would ask the participant to simply "choose which audio example is better described by the given descriptive term"—a simple task that requires comparing against one other example rather than all previously rated examples. After at least four individuals contribute the same descriptive term, we can estimate the quality scores for the descriptor using the Audio-Evaluator model, converting the binary comparisons into continuous ratings and estimating the reliability of each individual based on their consistency and hearing tests. With these continuous-valued ratings, we can use Sabin's equalization (EQ) learning algorithm (see Section 2.4) without modification to estimate a relative spectral curve that represents the aggregate opinion of the population.

We can then use this relative spectral curve to map to the parameter space and build equalization descriptor maps. With the relative spectral curve and a user-specified gain (a value that represents the strength of the applied effect, e.g., "How tinny do you want to make your sound?"), equalization parameters can be generated. To build a two-dimensional descriptor map for equalization, we can use multi-dimensional scaling (MDS) with a cosine distance matrix of relative spectral curve pairs, and we can size the descriptors on the map using the WST of the pairwise comparisons which can indicate how well defined the concept is.

3.9. Conclusion

In this chapter, I presented my work on communicating audio concepts to software with evaluative feedback. I proposed the first audio-specific annotator model for aggregating audio evaluations from a crowdsourced population of listeners in varied conditions to obtain "clean" data. We can use this aggregate clean data to learn agreed-upon mappings to audio concepts from input such as audio descriptors as we did in SocialEQ but without SocialEQ's limitations, and using such learned agreed-upon mappings, we can build affordances into audio production tools.

I investigated this from the more general lens of crowdsourced audio quality evaluation. In doing so, I developed recommendations and software for web-based, crowdsourced listening tests. These tests enable researchers to evaluate audio examples quickly and easily, estimating aggregate quality scores from a population of listeners. I evaluated two different types of listening tests on the web—web-MUSHRA and pairwise-comparison. I proposed simple procedures for estimating information about a participant's hearing response, listening environment, and reliability, and I proposed methods for incorporating this information into the estimation of population-based audio quality scores. I collected web-MUSHRA evaluations from 530 participants in only 8.2 hours and established that researchers could move MUSHRA to the web with minor modifications and obtain score estimates comparable to lab-based MUSHRA. However, MUSHRA is limited to 12 stimuli or less and requires ground-truth reference stimuli. Pairwise-comparison tests do not have these limitations, so I also crowdsourced a pairwise-comparison listening test. I recruited 458 participants for the pairwise-comparison test and found that web-based, crowdsourced pairwise-comparison listening tests can also estimate scores comparable to lab-based MUSHRA. I also found that our methods (e.g., the Audio-Evaluator model) to incorporate hearing response, listening environment, and reliability information made estimated scores from our web-based tests more similar to scores from the lab-based test for the *overall quality* scale. Lastly, I introduced the Crowdsourced Audio Quality Evaluation (CAQE—pronounced 'cake') Toolkit, a software package for running web-based, crowdsourced listening tests.

The methods presented in this chapter can be used to crowdsource the evaluation of all forms of audio: audio systems, audio algorithms, synthesizer sounds, and even creative output such as mixes or compositions. The aggregate quality scores estimated by these methods can also be used to train models. By asking a population of listeners to do a simple evaluative task such as rating or picking an audio example, we can learn general, high-level audio concepts that are agreed upon by a target population of listeners. This is similar to our goals with SocialEQ (see Chapter 2), but by aggregating evaluations instead of models, we can overcome SocialEQ's limitations. We can then use the learned high-level audio concepts to build affordances for the target population into tools for audio production, hearing aid tuning, audio search, etc.

CHAPTER 4

Vocal Imitation Part I: SynthAssist

4.1. Overview

In this chapter, I present research on communicating audio concepts to software using examples in the form of non-speech vocal imitations (e.g., the <*Bowubwubwub*>[1] in "give me a <*Bowubwubwub*> sound"). While the descriptive language approach from Chapter 2 works well for some audio concepts, vocal imitations are more effective for communicating audio concepts that are unidentifiable or synthesized abstract sounds. I propose a novel algorithm for mapping vocal imitations to audio concepts. This algorithm is incorporated into SynthAssist, a query-by-vocal-imitation system I developed for searching the space of a synthesizer. It is the first audio production interface with which users program a synthesizer using vocal imitations (i.e., "auditory sketches") and evaluative feedback. By doing so, the system provides affordances to novices that do not have the technical knowledge required of typical synthesizer interfaces. While developed to query synthesizer sounds and their associated parameters, SynthAssist's query-by-vocal-imitation algorithm also supports searching other databases of audio files such as sound effects libraries for sound editors / designers.

[1] Note that the written representation of vocal imitations (e.g., <*Bowubwubwub*>) are poor approximations to the actual sound of the vocal imitation and should not be considered equivalent.

The work described in this chapter was presented at the Conference on New Interfaces for Musical Expression [**20**], won the Best Demo Award at the ACM International Conference on Multimedia [**19**], and has recently been awarded a US Patent [**25**].

4.2. Introduction

Vocal imitation is a convenient, accessible method of audio concept communication that conveys an approximation of pitch, rhythm, loudness, *and timbre* using just one's voice. In the last chapter, I presented a method to improve the human-to-computer communication of audio concepts using quality evaluation by a population of listeners. However, audio concept communication using just audio evaluation is time consuming and can take several minutes for even simple audio-concepts [**156**] since audio is a temporal medium requiring time to listen to and evaluate each example. This is in contrast to evaluation-based image-concept learning systems like CueFlik [**49**] in which examples can be evaluated quickly "at a glance". In Chapter 2, I hasten novice human-to-computer communication of audio concepts with a method that uses a slow, offline evaluation-based method of audio-concept communication to learn mappings for a faster method of audio-concept communication—descriptive language. Descriptive language can work well for some audio concepts, but studies have shown that *vocal imitations* are more effective for communicating audio-concepts that are unidentifiable or synthesized abstract sounds [**103, 102, 104**]—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.

ATTORNEY: Could you describe for us Mr. Jackson the process and sequence of events which you use in composing songs generally, and then we'll get to "Dangerous" specifically. Is there a kind of process that you go through?

MICHAEL JACKSON: Well, usually when I write songs... I vocally... use melody into a tape recorder. And for instance with this song "Street Walker", which has a driving bass lick that I just spoke about... I have a tape a recorder and I just sing the bass part into the tape recorder. And for Street Walker, the bass melody went <*bhum-pah kah-ah pahhhh uh-ah bhum-pah kah-ah pahh-ah uh-ah bhum-pah kah-ah pahh-ah uh-ah*>. And I taped that bass lick and put the chords of the melody over the bass lick, and that's what inspires the melody or the other sounds that I'm hearing in my head...

Figure 4.1. A transcription excerpt from a 1994 deposition of Michael Jackson [**84**]

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, when observing a recording session, Porcello recalled how a novice recording engineer and an experienced engineer communicated various snare sounds 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*>" [**140**]. Such communication isn't limited to recording engineers though. In a 1994 court case, Michael Jackson, who was not proficient on any instrument and couldn't read or write music, describes how he sketches out all of his songs by recording vocal imitations of the instrumental parts (see Figure 4.1). Studio musicians would then listen to the imitations and record over them using their instruments [**84**]. Even formally educated composers

INNER VOICE 1: Yeah, let's get back to where we were. There's this nasal pulsing thing. Doesn't anyone remember that?
INNER VOICE 2: <*eee eee eee*>
INNER VOICE 1: ...and then suddenly there's going to be this kind of wind sound like
ALL INNER VOICES: <*wwhooooohhhoohshhoooooh*>
INNER VOICE 1: ...and then uhhh... ummm... I don't know. What do you think should happen next?
INNER VOICE 3: hmmmm
INNER VOICE 4: It could get like distorted and staticky like <*kwuuuucckckckkckeeeeh*>
INNER VOICE 3: Ohh yeah, I like that. That's cool.

Figure 4.2. A transcribed excerpt from Mark Appelbaum's *Pre-Composition* (2002) [**4**]

like Stanford professor Mark Appelbaum use vocal imitation to communicate composition ideas—as can be heard in his piece *Pre-Composition* (2002) [**4**] (see Figure 4.2), 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 [**41**].

Despite these promising examples, until recently, computational support of sketching in the auditory domain to aid in digital audio production and sound design has largely been ignored. This is in contrast to the popularity of sketching as an element of design processes [**16**] and the decades that researchers have spent decades developing computational tools to support visual sketching in digital design applications [**89**].

Audio synthesizers are a seemingly natural fit to interact using auditory sketching methods like vocal imitation. An audio synthesizer is an audio production and sound design tool for creating new sounds using sound generators such as oscillators, noise, and

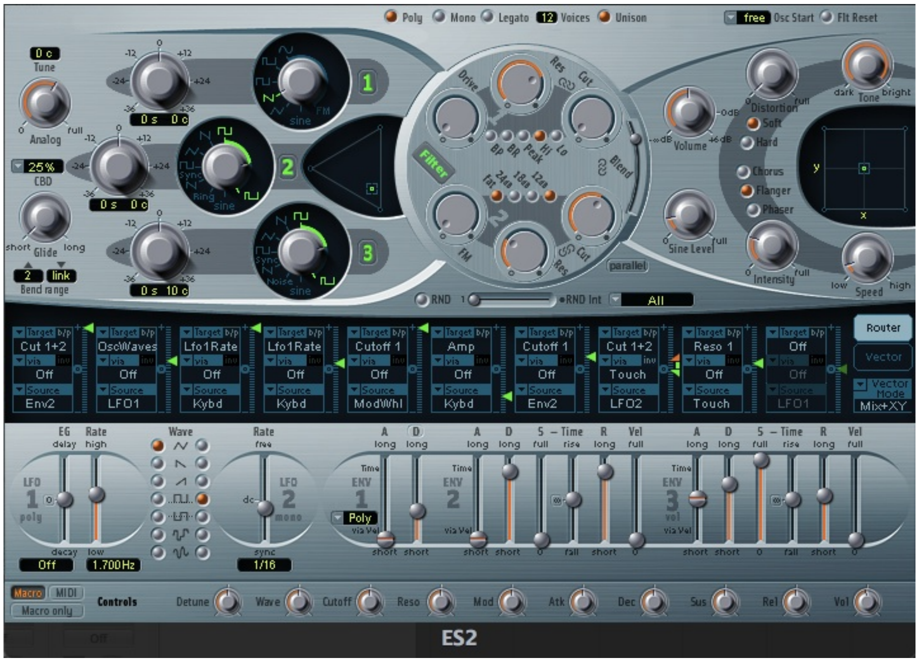


Figure 4.3. Screenshot of Apple's ES2, a typical software synthesizer. Introduced in 2001, it is still shipped with Apple's Logic Suite.

samples, which are further processed by filters, effects, and other processing units. Users typically communicate their desired audio concept (i.e, synthesized sound) to synthesizers using traditional synthesizer interfaces that consist of knobs, sliders, and buttons that control low-level synthesis parameters.

Audio synthesizers are also some of the most complicated audio production tools. (see Figure 4.3). For example, Apple Inc.'s ES2 synthesizer has 125 controls. If those controls were simply binary switches, the control space would consist of 2^{125} (i.e. 10^{38}) possible parameter combinations. Knobs and sliders of course have more settings and allow even more parameter combinations. These parameters also often interact with each other and may have non-linear relationships with our perception of the resulting sound. Fully exploring such a large space of options is difficult. Compounding this problem is the fact that controls often refer to parameters whose meanings are unknown to most (e.g. the ‘LFO1Asym’ parameter on Apple’s ES2 synth). Some manufacturer try overcome the barriers of complex traditional synthesizer interfaces by having many, many presets (e.g. Native Instruments Kore Browser). However, searching through a vast number of presets can be a task as daunting as using a complex synthesizer.

To overcome these barriers and provide affordances for novices, I have developed SynthAssist, an audio synthesizer interface that allows users to communicate their desired audio concept (a synthesizer sound in this scenario) to a synthesizer using vocal imitation and evaluative feedback. With SynthAssist, a user can quickly and easily search through thousands of synthesizer sounds. The user first provides one or more *soft examples*—e.g., example sounds that have some, but not all, of the characteristics of the desired sound. I have designed and tested SynthAssist with the motivation that these soft examples are vocal imitations, but the flexible representation that I use allows these soft-examples to also be existing recordings that are similar to the desired goal. Given these input examples, SynthAssist guides the user in an interactive refinement process, where the system presents suggested sounds for the user to rate. Based on these ratings, it learns what features are important to the user so that it can provide a sound that closely matches the user’s desired audio concept.

While this method has been specifically designed for synthesizer and musical instrument sounds, it could also potentially be used for other audio query-by-example (QBE) applications such as searching through a sample or sound effects database (e.g. finding the right door “slam” from a library of 100,000 sound effects). Sound effects retrieval systems typically require users to search using keywords queried against short descriptions stored with the audio. SynthAssist could provide an alternative search method that narrows the results based on desired audio characteristics communicated through vocal imitation.

4.3. Background

4.3.1. Other forms of vocal imitation and non-traditional vocal input

I am interested in non-conventional (i.e. not codified in language) vocal imitation that tries to mimic a referent sound as closely as possible—imitating its pitch, rhythm, loudness, and timbre. However, most previous research on voice input for audio/music software applications (e.g., music retrieval, sound retrieval, musical sequencing, and the control of digital musical instruments (DMIs)) has focused on more limited forms of vocal imitation and non-traditional vocal input.

For some applications, just vocal imitation of the pitch and rhythm of an audio or musical concept is used as input. For instance, query-by-humming (QBH) is a method for retrieving songs and musical works by humming or singing the melody with your voice [**79, 55, 132, 10**]. Such systems have primarily focused on matching queries to musical works based on pitch and the rhythm, ignoring timbre. In 2013, an audio-to-midi feature was added to the popular Ableton Live 9 [**1**], letting users sequence musical lines using the just the pitch and rhythm of their voice. Pitch and rhythm may be sufficient for communicating melodies, but when searching sound effects databases or programming synthesizers, timbre is a critical feature.

In other applciations, researchers have explored languaged-based vocal imitations of timbre—e.g., onomatopoeia—for sound and music queries. Onomatopoeia are language-dependent words that suggest a particular sound, e.g., “boom”, “splash”, “buzz”. Gillet and Richard [**57**] expanded on the idea of query-by-humming with a system of drum loop retrieval that utilizes spoken onomatopoeia sequences as queries, mapping a small a set of onomatopoeia to specific drums. Sundaram and Narayanan [**182**] explored using *written* onomatopoeia as queries to search a sound effects database where the sounds were represented in a onomatopoeia vector space generated from manual tags. While onomatopoeia do contain some information related to the timbral properties of sound, they are limited in number because they are dependent on language. Most sounds do not have associated onomatopoeia words. In contrast, general vocal imitation is only limited by what sounds someone can make with their voice.

Researchers have also investigated using the timbral properties of the voice to control and explore the timbral space of DMIs in real-time rather than to communicate a preconceived sound. In other words, users do not vocally imitate their desired sound to these applications, but they use the expressive, timbral properties of their voice to control a synthesis engine. Stowell, Fasciani, and Janer [**181, 46, 85**] have all capitalized on the natural, expressive quality of the voice by mapping properties of the voice to synthesizers controls for real-time gestural performance. Janer [**85**] mapped voice to control synthesizers using a fixed mapping, focusing on performance of pitch, dynamics, and articulations. Stowell [**181**] and Fasciani [**46**] both learned mappings between the timbre of the voice and the timbre of a synth.

While the approaches of [**181, 46, 85**] provide interesting ways of exploring the synthesis space, they require the user to use their voice to continuously control the dynamics of the synthesizer—in other words, they don't program the synthesizer; they perform it, requiring skills a novice may not have. For example, if a user has a preconceived desired sound, they would first have to vocally explore the space to the learn the mapping, then they would have to have excellent control over their voice to exactly perform their desired sound in reference to the mapping. In addition, in many scenarios, this is not feasible and/or desirable (e.g. the performer also sings, the microphone is picking up other loud instruments, etc.). Lastly, all of these projects also make certain assumptions about how people map features of the referent audio onto the vocal imitations.

4.3.2. Prior research on vocal imitation

There have been a few projects that have explored general non-language-based vocal imitation as input. Blancas and Janer [**12**] investigated vocal imitations queries for searching sound effects databases. They trained classifiers using hundreds of audio features. However, their study was limited to just a handful of audio classes. Smaragdis and Mysore [**174**] developed "separation-by-humming", a user-guided source separation technique in which vocal imitation is used to inform the priors of a probabilistic audio source separation algorithm and therefore select and separate the referent audio. However, their approach treated the voice as any other audio and did not address the limitations of the voice and how people imitate sounds.

Accounting for the limitations of the voice is one of the biggest challenges of effectively utilizing vocal imitations as input. The voice is not only limited in its pitch range and rhythmic precision, but also in its timbral range—the variety of sounds it can produce. When imitating sounds, we likely utilize different strategies to accommodate for these limitations.

Lemaitre et al have investigated vocal imitation from several angles [**36, 100, 102**] and have begun investigating these limitations and imitations strategies. In [**100**], participants freely categorize 720 total vocal imitations of 12 identifiable kitchen sound recordings (3 from each of the following categories: *gases*, *liquids*, *electrical*, *solids*). The aggregate clustering of the data agreed for the most part with the categories of the referent sounds, however imitations within each of the 4 categories often weren't matched with their referent sounds. They also showed that they could train a machine to classify both vocal imitations and their referent sounds using the same audio features. While a promising initial investigation into what sounds can be communicated to software via vocal imitation and with what features, Lemaitre speculates that participants may have adapted their imitation strategy to this small set of sounds, choosing features that would easily differentiate them, and therefore these results may not be generalizable.

Lemaitre et al also investigated both vocal imitation and descriptive language as a means of communicating audio concepts [**102**]. Using a set of 36 sounds (9 from each of the following categories: 1. *identifiable complex events*, 2. *elementary mechanical*, 3. *artificial sound effects*, 4. *unidentifiable mechanical sounds*), they found that vocal imitations were no more effective than descriptive language at communicating for referent sounds in which the source is identifiable (categories 1 and 2). However, vocal imitations were much more effective than descriptive language when the referent sound sources were not identifiable (categories 3 and 4).

They also began looking at the acoustic properties of effective and noneffective vocal imitations. They noted that imitations were most effective when they were able to successfully convey both the temporal and spectral information of the referent sound (e.g.

discrete noise bursts of a spray deodorant can), were moderately effective when if they conveyed a salient feature regardless of the other characteristics (e.g. a sine tone that sequentially stepped up in pitch was easily recognized regardless of the fact the timbre of the imitations were more complex than a single sine tone), and were least effective when neither the spectral or temporal information was conveyed (e.g. coins dropping that contain many overlapping events and resonant high frequencies). However, these results were primarily derived from a few examples in the experiment, and more general results would require a larger dataset.

4.3.3. Synthesizer Interfaces

Becase of their lack of affordnaces for novices, research related to the development of intuitive interfaces to audio synthesizers has been ongoing for several decades, with numerous approaches having been proposed [**9, 53, 86, 120, 200, 31, 45, 74, 114, 77, 191, 196, 85, 119, 66, 181, 88, 47**]. A general approach researchers have sought is to reduce the dimensionality of the synthesis parameter space by re-mapping controls to perceptual dimensions [**196, 191**], high-level descriptive dimensions [**45, 77, 88**], exploratory maps [**9, 120**], gestural spaces [**47**], other timbral spaces (e.g. the voice [**181, 46**], other instruments [**86**], etc.), and more.

Another approach is to explore the synthesis space by using an interactive genetic algorithm to optimize the synthesis parameters. Here, the user's judgment is the fitness function [**31, 114**]. The number of evaluations required to program a specific desired sound in this way is far too great to be completed by a human, but the approach can be useful for pure exploration.

In a related approach, the human is removed from the loop, and the fitness function is computed by a distance function and minimized in an optimization algorithm [**114, 199, 200, 74, 119, 53**]. In these "tone matching" approaches, the user provides an example audio file of the *exact* desired sound, and the tone-matching algorithm optimizes the parameters of the synthesizer to match the example audio file as closely as possible within its synthesis space. To achieve this, researchers have used many types of optimization procedures: genetic algorithms [**199, 200, 74**], interactive genetic algorithms (hybrid of computer-based and user-based objective evaluation) [**114**], genetic programming [**53**], linear programming [**119**], particle swarm optimization [**66**], and hill climbing [**200**].

An alternative to the optimization-based tone-matching methods is a "data-driven" approach proposed by Yee-King [**200**] that samples the parameter space of a synthesizer, producing thousands of example synthesizer settings. Descriptive features for each example are stored in a database, along with the synthesis parameters required to produce the sound. Given a recording as a query, the system simply finds the nearest neighbor in the feature space. When optimizing both a 22-parameter frequency modulation (FM) synthesizer and a 17-parameter subtractive synthesizer, they randomly sampled 100,000 parameter combinations and achieved comparable results to genetic algorithm and hill-climbing methods but with much faster retrieval times. While 100,000 sampled parameter combinations is only a very small fraction of the number of possible parameter combinations, they showed through "error-surface" analysis of the parameter space that the resulting feature space is very redundant.

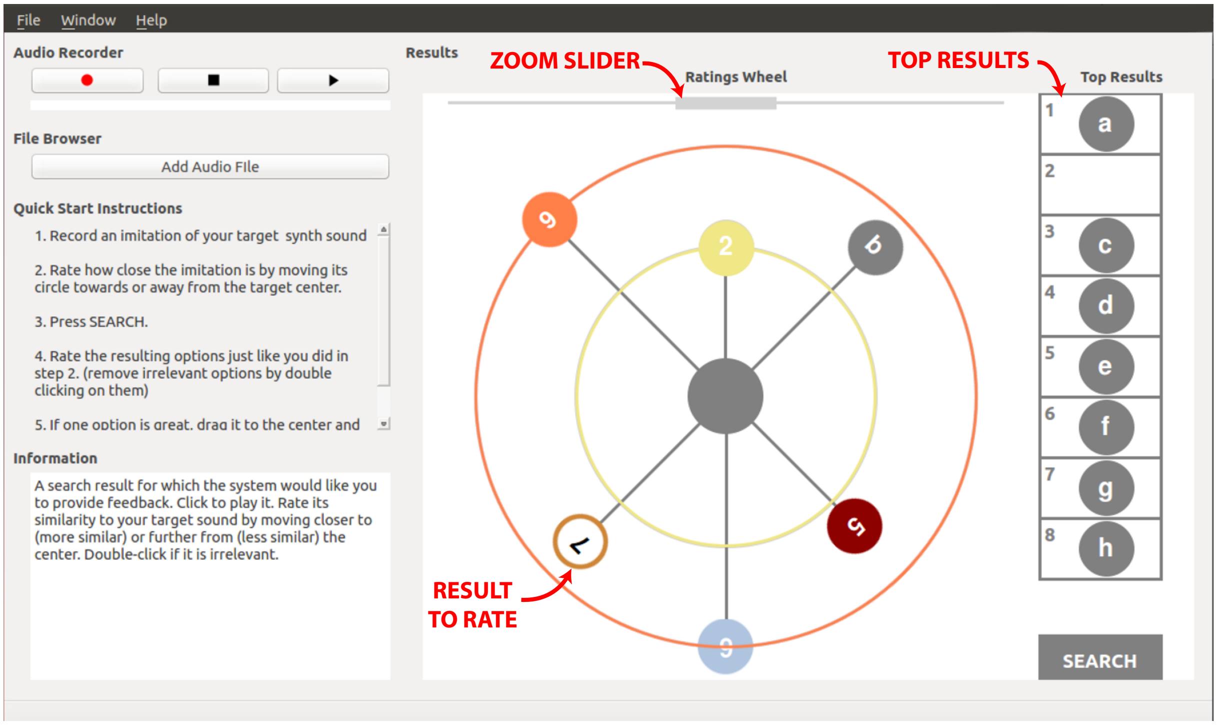
SynthAssist also employs a data-driven approach to synthesizer programming. However, in contrast to Yee-King's method for which the input must be a recording of the *exact* sound to be imitated by the synthesizer, our approach can be used with *inexact* examples or vocal imitations of the desired sound.

4.3.4. Audio Sample Retrieval

Both Yee-King's and our approach are essentially QBE audio retrieval systems in which the database contains both synthesized sounds and their associated synthesis parameters. QBE systems for audio retrieval have also been a rich research topic in the last 2 decades [**26, 39, 44, 67, 69, 68, 71, 94, 142, 198**]. It is one of many approaches to audio sample retrieval which also include methods for searching by semantic descriptors [**183**], clustering, and various visualization methods [**65, 39, 163**]. Within the audio QBE literature, researchers typically have modeled audio examples by the distribution of the frame-based features [**71**]. However, some methods retain the time series of each feature and directly compare the time series to calculate distance. Esling et al [**44**] take an interesting approach in which they individually compare each feature's time series, but rather than combine these distances into a single metric, they return the results that lie on the multi-objective Pareto front. At the end of that work they propose the idea of using the method in a "query-by-vocal-imitation" context by comparing the relative rather than absolute time series, but did not implement that. In this work, we do implement a query-by-vocal-imitation approach, combined with an interactive refinement process.

4.4. The SynthAssist System

SynthAssist is a system that allows users to program a synthesizer using vocal imitation and evaluative feedback. It uses a data-driven audio-retrieval approach in which



Quick Start Instructions

1. Record an imitation of your target synth sound
2. Rate how close the imitation is by moving its circle towards or away from the target center.
3. Press SEARCH.
4. Rate the resulting options just like you did in step 2. (remove irrelevant options by double clicking on them)
5. If one option is great, drag it to the center and

Information

A search result for which the system would like you to provide feedback. Click to play it. Rate its similarity to your target sound by moving closer to (more similar) or further from (less similar) the center. Double-click if it is irrelevant.

Figure 4.4. Screenshot of the SynthAssist interface.

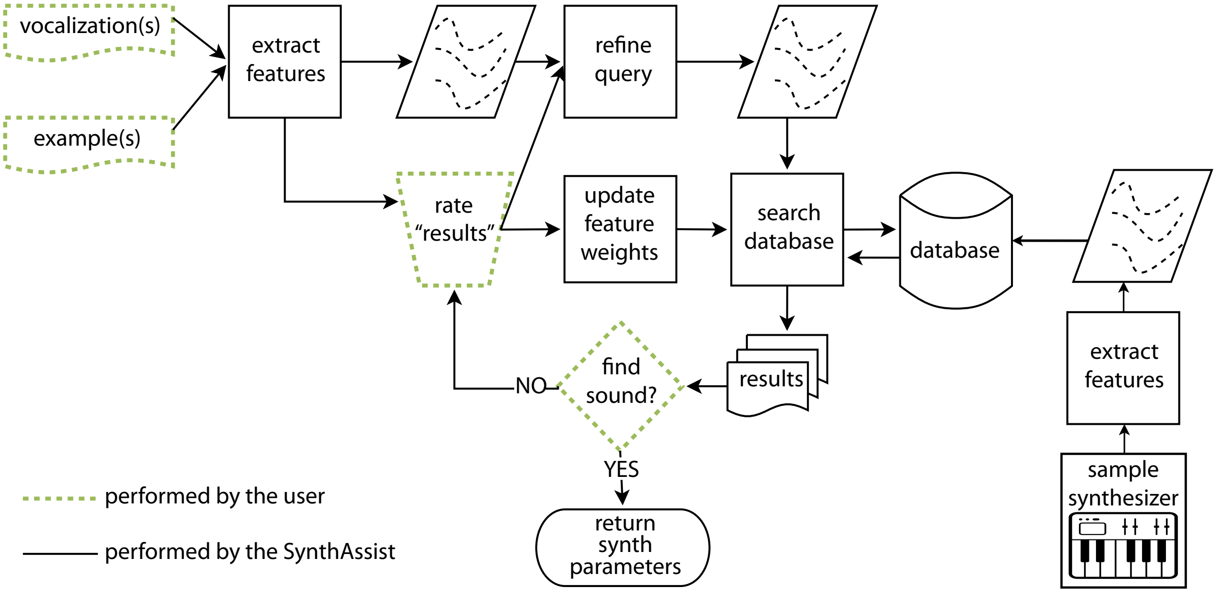


Figure 4.5. System flow of SynthAssist. The legend indicates which actions are performed by the user and which actions are performed by SynthAssist.

synthesized audio samples and their associated audio features and synthesis parameters are queried using vocal imitation and the query and the distance function are refined and adapted using evaluative feedback.

SynthAssist works as follows (see Figure 4.5 for an diagram of the system flow):

(1) **SYNTHASSIST:** First, SynthAssist randomly samples thousands of synthesizer settings from a particular synthesizer and models the sampled synthesizer settings by extracting time series of audio features. This first step is only performed once for each synthesizer it is configured for.

(2) **USER:** The system interaction begins when the user records their vocal imitation of their desired sound and rates how similar they believe their vocal imitation is to their desired sound.

(3) **SYNTHASSIST:** SynthAssist models the desired sound by extracting time series of audio features from the vocal imitation.

(4) **SYNTHASSIST:** SynthAssist compares the desired sound model to synthesizer setting models.

(5) **SYNTHASSIST:** SynthAssist suggests synthesizer settings—both synthesizer settings that the system believes are most similar (i.e., the top search results) to the desired sound and synthesizer settings the system wants the user to rate (a diverse set of search results).

(6) **USER:** The user listens to the suggested synthesizer settings. If the desired sound is close enough to a suggested setting, the user terminates the search process and uses the synthesizer with the chosen setting. Otherwise, the user

evaluates the suggested setting by rating how similar they are to the desired sound.

(7) **SYNTHASSIST:** Based on evaluative feedback, SynthAssist refines its model of desired sound and adapts the similarity function by re-weighting the audio features.

(8) **SYNTHASSIST:** Go to step 4 (i.e., repeat until the desired sound is found)

Figure 4.4 shows a screenshot of SynthAssist. Each synthesizer setting suggestion is represented by one of the colored circles. When a user clicks on a suggestion, an example sound plays and the synthesizer is configured to the setting. Users rate how similar suggestions are to their desired sound by moving the circles closer to or farther from the center, "hub", circle. If the suggestion is irrelevant, the user can inform SynthAssist and remove it from the screen by double clicking on it. Dragging a suggestion to the center of the circle indicates that it is the desired sound and terminates the interaction.

4.4.1. Modeling the Desired Sound and Synthesizer Settings

In SynthAssist, a single database entry contains the 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. The desired sound model is used to query to this database.

In SynthAssist, we want to support vocal imitation as input to the system. Our approach is similar to the approach suggested in [**44**]. The motivation is that while our voice has a limited pitch range and timbral range [**181**], 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.).

Focusing on these changes through time, we model both the synthesizer settings and the desired sound by extracting the time series of a small number of high-level features from the audio[2] (vocal imitations and example audio recordings of synthesizer settings): *pitch*, *loudness*, *inharmonicity*, *clarity*, *spectral centroid*, *spectral spread*, and *spectral kurtosis*. Definitions for all of the features but the *clarity* measure can be found in [**137**]. Similar to autocorrelation height, the clarity measure is a measure of how "coherent a note sound is" [**116**]. We then augment this representation by also standardizing each of these features with themselves to capture the relative changes through time that are invariant to the scaling and offset of feature values (e.g. $\mathbf{x}_{std} = \dfrac{\mathbf{x} - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}}$ where $\mathbf{x}$ is the time series and $\mu_{\mathbf{x}}$ and $\sigma_{\mathbf{x}}$ are the mean and variance of $\mathbf{x}$). By standardizing these features, synthesizer setting models with similarly *shaped* time series to the desired sound model can also rank highly. For example, your voice may not be in the same pitch range of your desired sound, but you may be able to change the pitch of your voice similarly in time, but an octave lower—and therefore the relative time series may have a similar shape.

Therefore, we model the desired sound and synthesizer settings as 14 time series, one per feature: 7 "absolute features" and 7 "relative features". While many QBE systems characterize these time series as either distributions (e.g. modeling them with a Gaussian Mixture Model) or extract statistics and features (e.g. mean, variance, slope, modulation), we retain the time series representation to capture the temporal evolution of each sound.

[2] Note that features for the search keys are extracted using a frame size of 1024 and a hop size of 512 at a sample rate of 32 kHz, and that before feature extraction, all audio (queries and samples in the database) is summed to mono and RMS-normalized.

Each model is a matrix in which each feature time series is a row of length N, where N is the number of feature frames, e.g. $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{14}]^T \in \mathbb{R}^{14 \times N}$.

4.4.2. Rank Calculation

To search the database we calculate the distance from the desired sound model (the query) to each synthesizer setting model (the database keys). However, the time series in the desired sound model and the synthesizer setting model may not be the same length. Therefore, we use a distance measure based on dynamic time warping (DTW) [**125**] and treat each feature time series independently. We calculate distance using the following equation:

$$D_{\mathbf{X},\mathbf{Y}} = \sum_{i=1}^{14} w_i DTW(\mathbf{x}_i, \mathbf{y}_i) \tag{4.1}$$

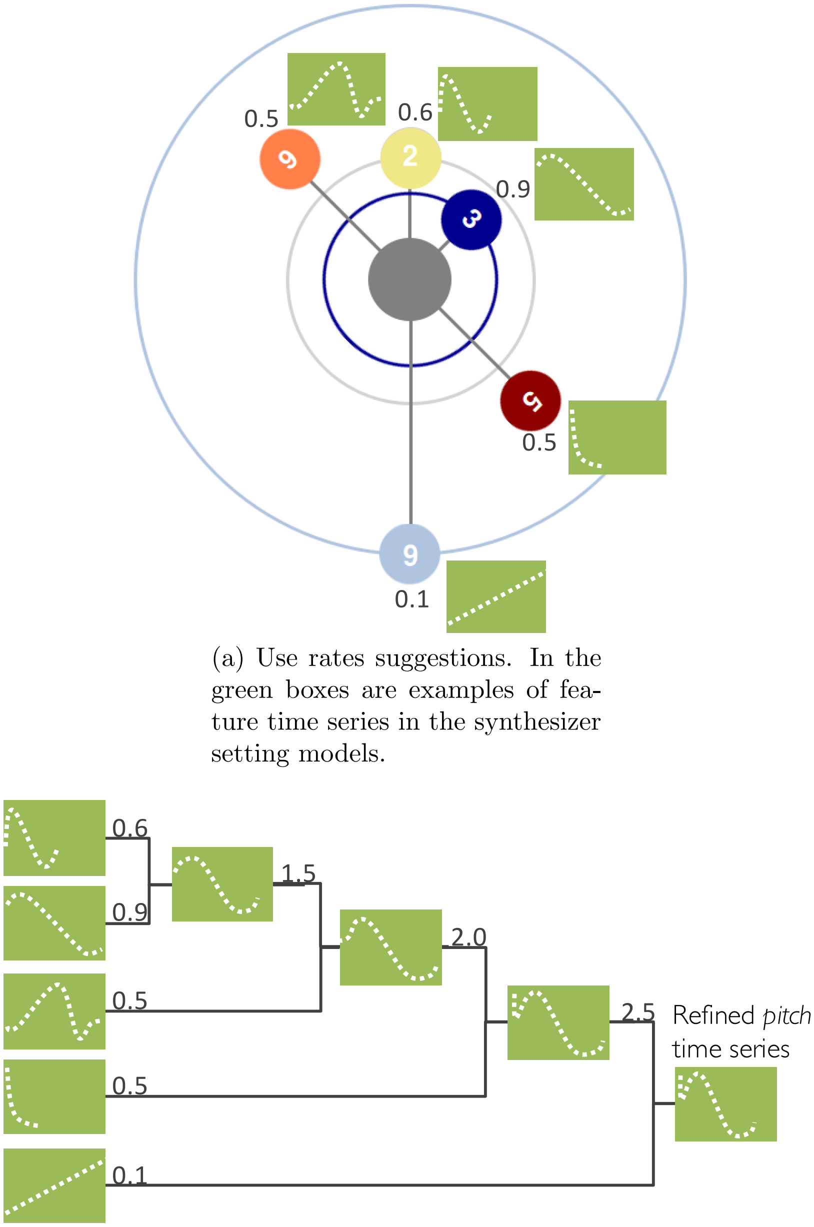
where i is the index of the 14 features, $\mathbf{X}$ and $\mathbf{Y}$ are the respective matrices of the desired sound and synthesizer setting models, $\mathbf{x}_i$ and $\mathbf{y}_i$ are the time series of the i^{th} feature for the desired sound and synthesizer setting models, w_i is the weighting coefficient of the i^{th} feature, and $DTW(\mathbf{x}_i, \mathbf{y}_i)$ is the DTW [**125**] distance between $\mathbf{x}_i$ and $\mathbf{y}_i$.

After calculating distance between the desired sound model and every synthesizer setting model, the system suggests two sets of results to the user: *top results* and *rating results.* The first set, *top results*, consists of the 8 nearest neighbors in increasing order of distance. Initially, the feature weighting coefficients are all equal, but after the first search round, they are refined as specified in the next section. Therefore, the top results may change each round (where a round is steps 4 to 7 in Section 4.4). This set of results appears on the right hand side of the interface as the array of grey circles as shown in Figure 4.4.

Since the top results may consist of many similar items, rating all of them may not give us much useful information. Therefore, the second set of results, *rating results*, consists of the example synthesizer settings we want the user to rate. This includes the nearest synthesizer setting that has not been rated yet and also the synthesizer settings which are nearest in each of the 14 feature dimensions (i.e. the keys that were closest while just considering each feature distance, $DTW(\mathbf{x}_i, \mathbf{y}_i)$, independently). This is a computationally efficient way of increasing the diversity of the results while also maintaining relevance. To avoid crowding the screen with too many results, we randomly select 7 of these 14 synthesizer settings. The *rating results* appear as the small colored circles radiating out of the "target hub" in Figure 4.4.

4.4.3. Relevance Feedback and Query Refinement

Once presented with the two result sets, the user can listen to both sets and can potentially select their desired synthesizer setting which also configures the parameters of the synthesizer. If their desired synthesizer setting is not in the results however, the user can give feedback to the system to improve the search. To give feedback, the user marks which of the *rating results* are irrelevant (by double clicking to remove the results) and rates each remaining relevant result by moving it closer to the center (more relevant) or further from the center (less relevant). All relevant results are added to the relevant set, Z. The relevant set includes all the relevant synthesizer settings for the current search session and the initial examples provided by the user.



(a) Use rates suggestions. In the green boxes are examples of feature time series in the synthesizer setting models.

(b) Treating the user similarity ratings as weights, SynthAssist uses Prioritized Shape Averaging to create a weighted average of feature time series of relevant synthesizer settings. These weighted averaged time series define the refined desired sound model.

Figure 4.6. Refining the desired sound model.

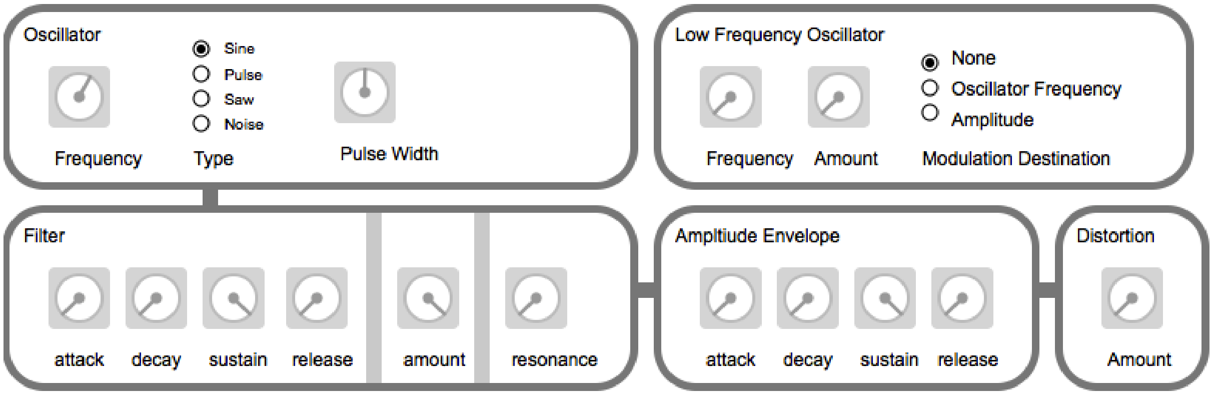
To encourage the user to scale their ratings consistently between sets of results, the user can see the position of the last round of result ratings in the form of light grey circles. The slider at the top of the screen lets users zoom in or out to scale their rating appropriately.

We convert these distance ratings to similarity ratings with the following function: $s_k = \dfrac{1}{d_k + 1}$, where s_k and d_k are the respective similarity and user-specified distance of the k^{th} relevant synthesizer setting. These ratings are used to get a more accurate estimate of what the user's ideal query is. These ratings are used to refine both the desired sound model and the distance function to more closely match the user's unknown ideal sound and internal similarity measure.

Our initial estimate of the desired sound model is given by the user's vocal imitation. We then refine our desired sound model (i.e., the query) by creating a weighted average of the feature time series of relevant synthesizer setting models. The weights are based on the user-provided ratings of the relevant synthesizer setting. In combining these, we are dealing with time series that are of potentially different lengths, since different synthesizer parameter settings can result in sounds of different lengths. In addition, we want to preserve the structure of these time series when we average them. For instance, imagine we have two time series based on the sine function, but is 180 degrees out of phase with the other. If these were given equal weight and added together, they would completely cancel each other out. However, we would likely perceive two examples as very similar if their only difference was that their feature modulations were out of phase. Therefore, we would like to preserve this structure in their time series.

DTW [**125**] can help us out in this scenario. We adopt the *prioritized shape averaging (PSA)* presented in [**126**] to create a weighted average of the time series. To average two time series, weighting in both time and amplitude, this method first finds the alignment path between the two time series via DTW. It then modifies this alignment path, warping it to make the resulting time series more similar to the time series with more weight. Using the indices from the modified alignment path, the elements from the two time series are indexed and combined using the weighted mean. To average more than two time series, this method first performs agglomerative clustering (a type of hierarchical clustering that results in a binary tree) on the time series, using DTW as the distance function. It then aligns (using DTW) and averages pairs of time series from the bottom of the tree on up, weighted according to the user-provided rating. Treating each feature independently, we use the PSA method to average the relevant time series for each feature. The resulting weighted average is our new estimate of the target audio concept, the refined desired sound model $\bar{\mathbf{X}}$.

In addition to refining the desired sound model, we also refine our distance metric in response to the user's relevance feedback. Recall that each of the example synthesizer settings presented to the user for rating is the closest one to the desired sound model along one of the 14 feature dimensions. To refine the weight w applied to each of the 14 features we use a simple inverse standard deviation relevance feedback mechanism, similar to those in the MARS and MindReader (when constrained to weight each feature dimension independently) image retrieval systems [**154**]. However, since we are dealing with time series, we calculate distance in the weighted variance function using DTW



● Sine
○ Pulse
○ Saw
○ Noise

● None
○ Oscillator Frequency
○ Amplitude

Figure 4.7. Screenshot of the traditional interface used in the experiment.

rather than the difference function. The calculation is as follows:

$$w_i = \left(\frac{1}{\sum_{k=1}^{|Z|} s_k} \sum_{k=1}^{|Z|} s_k DTW(\mathbf{y}_i^k, \bar{\mathbf{x}}_i)^2 \right)^{-\frac{1}{2}} \tag{4.2}$$

where w_i is the weight of the i^{th} feature, s_k is the user's similarity rating, $\mathbf{y}_i^k$ is the time series of the i^{th} feature of the k^{th} relevant synthesizer setting, and $\bar{\mathbf{x}}_i$ is the time series of the i^{th} feature of the refined desired sound model.

4.5. User Study

To evaluate this system's interaction paradigm and affordances for novices, we ran a user study with 16 participants who were pre-screened to have minimal experience with synthesizers. In the study, we compared SynthAssist to a traditional synthesizer interface consisting of knobs and buttons (see Figure 4.7). We did so by asking the participants to use the interfaces to match the synthesizer's output to target sounds. For this study, we used a 15 parameter synthesizer. This synthesizer had a simple architecture which consisted of one oscillator (selectable as sine, pulse, saw, or noise) which could be amplitude or frequency modulated by a low frequency oscillator (0–200Hz). The oscillator was fed into a envelope-controlled filter stage, then to an envelope-controlled amplifier stage, and lastly to a distortion stage.

We generated 10,000 synthesis settings that were selected to evenly cover the parameter space of the synthesizer and added them to SynthAssist's database. The participants were given four target audio examples chosen from the database. The participants were then asked to use each synthesizer interface to match the sounds as closely as possible. While examples could be listened to as often as the user desired, they could not be used as input to SynthAssist. The initial query provided to SynthAssist was limited to the user's vocal imitation of the target audio example. We counterbalanced the order of the interfaces and stimuli using a Latin squares design.

Before using each interface, the experiment administrator spent a few minutes explaining and walking the participants through the interface. They were given five minutes to match each sound. In addition, at each minute mark, they were asked to rate how close they perceived their best sound thus far was to the target sound on a continuous scale marked at eleven, evenly spaced levels with "very dissimilar", "neutral", and "very similar" marked at the extremes and middle. Each interface was completely reset for each new target. After trying to match the four target sounds with each interface, the participants were asked to fill out a survey about their experiences with each interface.

According to the survey, the mean age of the participants was 27.8 (SD=5.4) years, and there were an equal number of male and female participants. When asked to "Estimate the number of hours per week you actively use audio synthesis technology", the median answer was 'Less than one hour per week". However, when asked to "Estimate in years

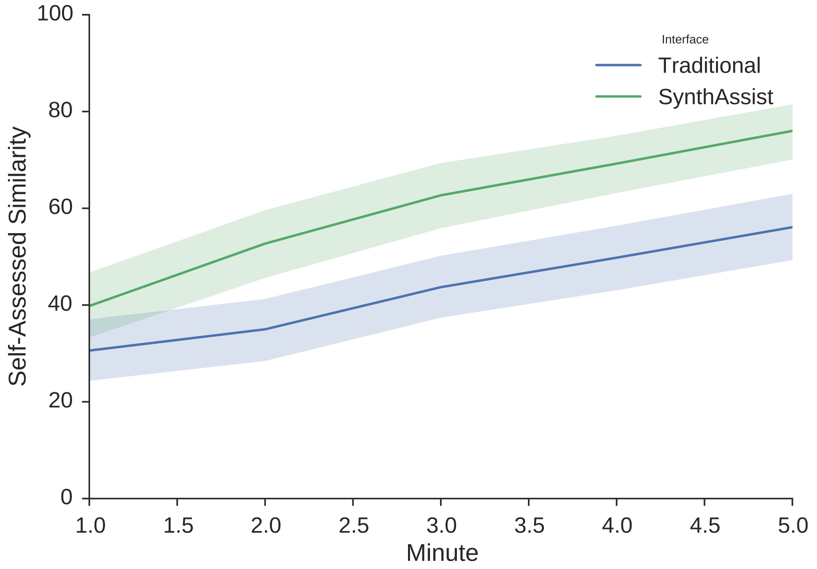
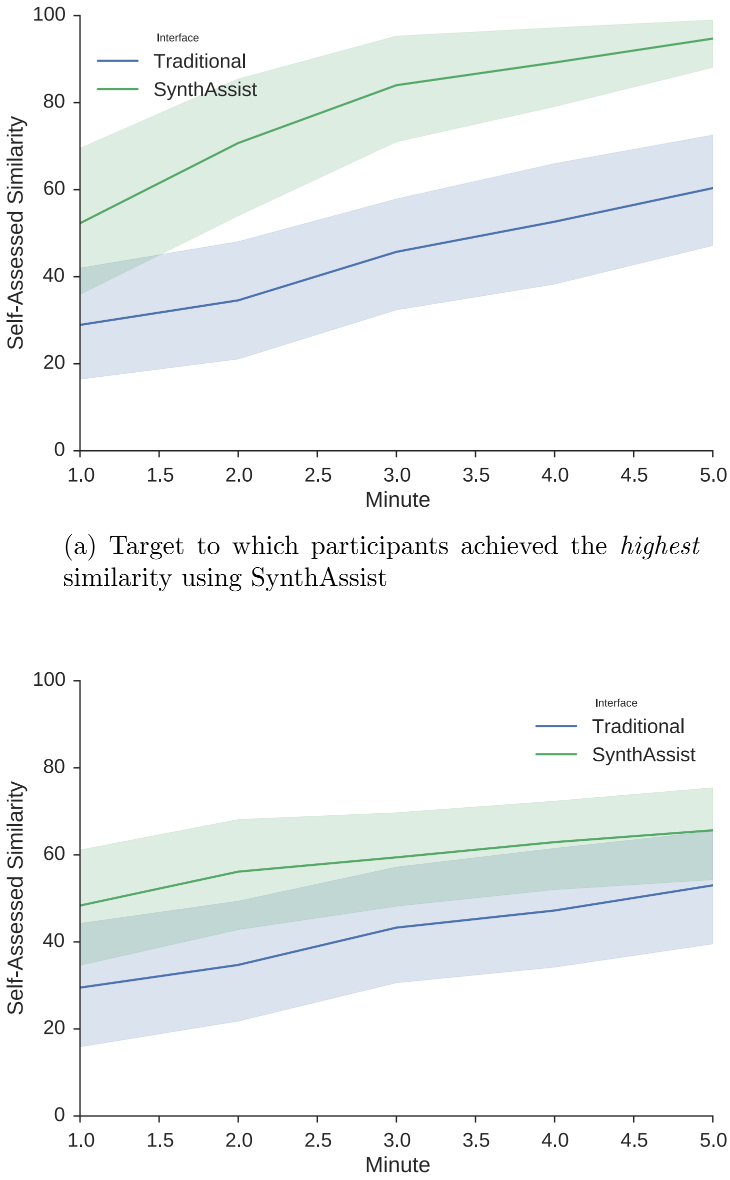


Figure 4.8. The one minute-spaced similarity evaluations between the given target sound and the most similar sound that a participant achieved in the timed trial thus far. On the y-axis, 0='very dissimilar', 100='very similar'. The plotted lines are the means of all participants for all targets. The bands are the 95% confidence intervals (CIs). N=64.

how long you have been actively using audio synthesis technology", the mean was 1.9 (SD=3.4) years. Therefore, a few participants did have some experience with synthesizers despite the pre-screening.

The overall self-assessed perceptual similarity results are shown in Figure 4.8. 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. On average, it took participants three minutes with the traditional interface to obtain sounds of an equivalent target similarity as SynthAssist obtained after one minute.



(a) Target to which participants achieved the *highest* similarity using SynthAssist

(b) Target to which participants achieved the *lowest* similarity using SynthAssist

Figure 4.9. The one minute-spaced distance evaluations of the target to which participants achieved the highest and lowest similarity using SynthAssist. On the y-axis, 0='very dissimilar', 100='very similar'. The plotted lines are the means. The bands are the 95% CIs. N=16.

We also asked participants survey questions about their experience with the two interfaces and summarized their responses by taking the median response of a 7-level Likert scale (*strongly disagree*, *disagree*, *somewhat disagree*, *neutral*, *somewhat agree*, *agree*, *strongly agree*). In reference to SynthAssist, participants *somewhat agreed* with the statement "It was easy to make the sound that I wanted", but *disagreed* in reference to the traditional interface. In reference to SynthAssist, they *somewhat disagreed* with the statement "Using this software was frustrating", but *agreed* in reference to the traditional interface. However, when asked to respond to the statement "The software did not always do what I wanted", they *agreed* in reference to the traditional interface and also *somewhat agreed* in reference to SynthAssist. Despite this however, in reference to traditional interface, they *strongly disagreed* with the statement "I would recommend using this software to computer novices", but *somewhat agreed* in reference to SynthAssist. Therefore, 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.

In fact, when we inspect the self-assessed perceptual similarity results for individual targets, we find the participants found some targets much easier to match with SynthAssist than others. However, the perceptual similarity results for the traditional interfaces are very similar across targets. In Figure 4.9, the perceptual similarity results are plotted for the target to which participants achieved the highest and lowest similarity. The target to which participants achieved the highest similarity was a noisy, resonant, percussive sound, while the target to which participants achieved the lowest similarity was a rich, harmonic, and highly modulated sound.

4.6. Conclusions

In this chapter, I presented research on communicating audio concepts to software with vocal imitations. I proposed a novel method for mapping vocal imitations to audio concepts. This query-by-vocal-imitation method can be used to search databases of audio for sound design such as sampled synthesized sounds or sound effects libraries, making audio that may be difficult to describe in words accessible to creators. This method is at the core of SynthAssist, a query-by-vocal-imitation system I developed for searching the space of a synthesizer.

SynthAssist represents a fundamental reinvention of the interaction paradigm of synthesizers. It is the first synthesis tool to which users can communicate audio concepts using vocal imitations. Using this system, a user simply needs to know what sound they are seeking, give it an initial example (e.g. using their voice to imitate the target), and be able to rate how similar example sounds are to their target sound. A user study showed that for novices this system is a promising alternative to traditional synthesizer interfaces and could empower user populations to effectively use synthesizers who otherwise were unable to.

In the user study, I also discovered that some audio concepts were much more easily communicated to the system than others using vocal imitation. Which leads us to the questions: *What makes some audio concepts easier to communicate to software than others? Are some sounds hard to imitate? Are some imitations easier to detect? Can we* *predict which audio concepts are communicable to software using vocal imitation?* The answers to these questions will inform us as to what applications vocal imitation input can be used for. In the following chapter, I present VocalSketch, a project aimed to answer these questions.

CHAPTER 5

Vocal Imitation Part II: VocalSketch

5.1. Overview

In this chapter, I present additional research on communicating audio concepts to software using examples in the form of non-speech vocal imitations. I present VocalSketch, a project which investigates which audio concepts people can effectively communicate with vocal imitation. In this project, hundreds of participants contributed over 10,000 vocal imitations and hundreds of participants also described and identified these vocal imitations, creating the first comprehensive and largest database of publicly available vocal imitations. This dataset helps us and the research community investigate the mapping between vocal imitations and the referent audio (e.g., how we accommodate the limitations of the voice when imitating sounds). This can help us understand which audio concepts can be effectively communicated with vocal imitation and what the characteristics of these audio concepts are. The dataset provides training data and a human performance baseline for query-by-vocal-imitation systems, and the dataset will inform us as to which applications can take advantage of this intuitive form of "auditory sketching".

The work described in this chapter won a Best of CHI Honorable Mention at the ACM Conference on Human Factors in Computing Systems [**22**].

5.2. Introduction

In Chapter 4, I proposed vocal imitation as an alternative method of communicating audio concepts to software, enabling novice users to overcome the technical barriers of audio production tools. However, while vocal imitation may be a natural way to communicate audio concepts [**102**], vocal imitations are often only approximations of a desired sound. This is because the human voice is limited in the sounds it can produce. People approximate a desired sound by mapping its pitch, timbre, and temporal properties to properties of the voice. However, the timbral range of the voice is limited by the physical acoustics of the vocal tract and vocal folds, and the vocal folds also limit the range of fundamental frequencies the voice can produce, typically in the range of 85-255 Hz [**187**]. Therefore, vocal imitation may be more effective for communicating some sounds than it is for others.

For vocal imitation to be recognized as a useful input method, it is important for interface designers to understand what kinds of sounds may be successfully communicated by vocal imitation so that appropriate applications can be identified and appropriate interfaces can be built. In order for those interfaces to function properly, it is also important to understand how people imitate sounds—how they map features of the referent audio onto features of their own voice. Lastly, when designing software such as a query-by-vocal-imitation systems, it is also important to have a human baseline to which we can compare retrieval performance. These are the concerns we seek to address with VocalSketch.

In the visual domain, Eitz et al addressed similar concerns by crowdsourcing thousands of visual sketches of 250 everyday objects [**40**]. The authors of that study created two tasks on Amazon's Mechanical Turk (AMT): a sketching task and a recognition task, and they also released their data set to the public for others to use. Inspired by Eitz's work, VocalSketch seeks to obtain a similar dataset for "sketching" in the auditory domain, i.e., vocal imitation. With VocalSketch, we collected thousands of labels and "vocal sketches" from participants over AMT.

5.3. Methodology

In this work, we focus on vocal imitation to communicate audio concepts defined primarily by their timbre. We assembled a diverse set of 240 audio concepts with a wide range of timbres. Most audio concepts were defined by both a sound label (e.g. a 1–4 word description, e.g. "barking dog") and a short (up to 10 seconds) sound recording of a specific instance of the audio concept. 40 of the audio concepts did not have sound labels (discussed later). We then collected thousands of vocal imitations from workers on AMT. A second set of workers described the vocal imitations and matched them to their referent audio concept. All workers were first required to pass a simple listening test.

This data set will help address the following questions:

(1) What types of audio concepts can be effectively communicated between people via vocal imitation?
(2) What are acoustic characteristics of audio concepts that can be effectively communicated between people?

The answers to these questions will clarify what software applications this method of communication can be used for. The human-provided labels of the vocal imitations form a performance baseline for automated systems that search for relevant audio based on vocal imitations.

These questions are similar to those that Lemaitre [**102**] sought to answer. However, one of the drawbacks of Lemaitre's work was that results were speculated to be idiosyncratic to the dataset and not generalizable. This was due to the small scale of the experiment. We overcome the problem of scale by crowd-sourcing the recording of vocal imitations over the internet rather than recording the vocal imitations in our laboratory. With our much larger data set, we hope to obtain more generalizable results.

When crowdsourcing listening tests, participants may be listening in a variety of listening environments, and while the human voice has limited frequency range, sound recordings in our audio concept set may have both very low frequencies and very high frequencies. To ensure that a participant's listening environment was adequate to produce a large range of frequency content, all study participants were required to complete a web-based listening test before participating in the study. This listening test was the same as the "hearing screening" described in Section 3.4.1.

5.3.1. Audio Concept Set

Our audio concept set contains four subsets: *everyday*, *acoustic instruments*, *commercial synthesizers*, and *single synthesizer*. These subsets were chosen with two goals in mind: 1) diversity and 2) applicability to potential applications for vocal imitation (e.g. sound design). Therefore, these subsets include everyday sounds as well as musical sounds that one might expect to find in audio applications such as software synthesizers, samplers, and sound effects search engines. See Table 5.1 for a list of the audio concepts.

The ***everyday*** subset is a set of 120 audio concepts assembled by Marcell et al [**110**] for confrontation naming applications in cognitive psychology (i.e. applications in which participants are asked to identify sounds). This set contains a wide variety of acoustic events of varying lengths (0.1 to 6 seconds) such as sounds produced by animals, people, musical instruments, tools, signals, and liquids. The set includes a recording and label (e.g. “brushing teeth”, “church bells”, “jackhammer”) for each of the audio concepts. The source publication of these sounds also includes guidelines for scoring the identification of each sound, and data from experiments that includes measures such as the complexity and pleasantness of the sounds.

The ***acoustic instruments*** audio concept subset consists of 40 primarily orchestral instruments [**52**]. Each sound recording in this subset is of a single note played on musical pitch C (where applicable) at on octave chosen to be appropriate for the range of each particular instrument. Where possible the note was sustained for several seconds. Longer sustained notes were edited down to a maximum of 10 seconds. The sound labels for the audio concepts are instrument names and any short notes on the playing technique (e.g. “plucked violin”).

The ***commercial synthesizers*** subset consists of 40 recordings of a variety of synthesizers in Apple Inc’s Logic Pro music production suite with various popular synthesis methods. This let us explore people’s ability to recognize and reproduce sounds that they could not necessarily name and had not had many years of exposure to (unlike “brushing teeth”). Each recording was created from a “factory preset” (well-crafted settings chosen and named by Apple Inc) and consists of a single note played on musical pitch C (the octave varied according to the intended pitch range of the preset) for four seconds. This length does not included the ‘release’ (i.e. tail) of the note. We set the maximum recording length to 10 seconds, therefore if the release was initially longer than six seconds, it was trimmed and gracefully faded to keep within the maximum length. The labels for this subset are the names of the factory presets, (e.g. “resonant clouds”, “abyss of despair”, “freaky moonlight”).

The ***single synthesizer*** subset consists of 40 recordings of a single 15-parameter subtractive synthesizer (with some limited FM and AM capabilities). Each recording consists of a note played on musical pitch C (the octave varied depending on the parameter settings) for four seconds with an additional maximum release time of three seconds. The synthesizer settings for each of the 40 recordings were chosen by the authors based on their diversity from a large set of randomly generated settings. This subset was included because we know the parameter settings used, and we have the source code for this synth. This data could be used to learn mappings between vocal imitation features, referent audio features, and synthesis parameters, which is of use to researchers of new music synthesis tools and interfaces. Since these recordings were not derived from presets and are difficult to describe with language, no labels exist for these sound recordings.

5.3.2. Preparation of sound recordings

All of the sound recordings were saved as mono, 16-bit .WAV files at a sampling rate of 44.1kHz. Note, the set of everyday sounds were originally at 22.05kHz and were upsampled to 44.1 khz for consistency with the other subsets. Sounds were trimmed at a threshold of -90dB to remove silence, and the whole set was normalized to be of similar loudness, using Root Mean Squared amplitude normalization.

Table 5.1. Labels of the audio concepts

Everyday (120 sound recordings, 120 sound labels)			
accordion	cow	horse galloping	shuffling cards
airplane	crickets	jackhammer	sneeze
baby crying	crow	knocking	snoring
bagpipes	crumpling paper	laughing	sonar
banjo	cuckoo clock	lawn mower	stapler
basketball	cutting paper	lion	swords
bicycle bell	cymbals	machine gun	tea kettle
birds chirping	dog barking	monkey	tearing paper
blinds	donkey	mosquito	telephone
blowing nose	door closing	motorcycle	thunder
boat horn	doorbell	ocean	toilet flushing
bongos	drill	organ	train
bowling	dropping ice in glass	owl	truck
brushing teeth	drums	piano	trumpet
burp	duck	pig	turning pages
camera	elephant	pinball	typewriter (manual)
can crush	explosion	ping-pong	velcro
can opening	firecrackers	police siren	violin
car crash	flute	pouring water	water bubbling
car horn	frog	pullchain lightswitch	water draining
cash register	frying food	rain	water dripping
cat	gargling	rattlesnake	whip
chewing	glass breaking	river	whistle (instrument)
chickens	gong	rooster	whistling (lips)
child coughing	guitar	sandpaper	wind
church bells	gunshots	sawing	wind chimes
clapping	hammering	saxophone	wolf
clearing throat	harmonica	scream	woodpecker
coin dropping	harp	seal	yawning
cork popping	helicopter	sheep	zipper
Acoustic instrument (40 sound recordings, 40 sound labels)			
piano	triangle	bass (bowed)	horn
violin (plucked)	trombone	bass (plucked)	thaigong
windgong	trumpet	bassoon	alto sax
woodblock	tuba	castanet	crotale
xylophone	vibraphone (bowed)	cello (bowed)	flute
oboe	vibraphone (sustained)	cello (plucked)	guiro
orchestra bells	viola (bowed)	clave	guitar
tambourine	viola (plucked)	clarinet	hihat
tambourine (shake roll)	violin (bowed)	crash cymbal (bowed)	crash cymbal (choked)
crash cymbal (hit with a stick)	crash cymbal (mallet roll)	chinese cymbal (hit with a stick)	marimba (hit with a rubber mallet)
Commercial synthesized (40 sound recordings, 40 sound labels)			
narrow timbre	spacecraft engine	bell swarm	filter bubble
needle strings	sparkle motion	breathing cyborg	freaky moonlight
orbit station	synth evolving lead	bullet time	ghost in the machine
photon blast	synth metallic stars	classic synth brass	ice synth
resonant clouds	thick bass	dark attack bass	little summer boy
reversed envelope	abyss of despair	dark synth pad	marble in a glass bowl
shadowland highs	air machine	dissonant bells	marble on a journey
shimmer	alien discovery	electrical interferences	metaloid
slow warm swells	anti matter clouds	euro hook	mister frosty
soft shimmer	bamboo air strings	feedback	morphing can drum
Simple synthesized (40 sound recordings, 0 sound labels)			
N/A (no sound labels for this subset)			

5.4. Vocal Imitations of Audio Concepts

We designed two tasks for AMT in which participants recorded a vocal imitation in response to a stimulus.

The first task addressed the use case where a user seeks a general audio concept (e.g. any church bells). Participants were given a *sound label* (e.g. the text "church bells"—see from Table 5.1 for a complete list of labels) from our audio concept set and asked to "imagine a short (less than 10 seconds) sound produced by the following description." Next they were asked to "indicate the degree of confidence you have in your ability to imagine the sound" on a discrete scale. They were then given a simple recording interface and asked to "imitate the imagined sound with your voice as closely as possible." They were told to avoid using conventional onomatopoeia (e.g. "meow"), but that whistles, coughs, clicks, and other mouth noises were okay. Before continuing to the next step they were required to listen again to their recording and to indicate how satisfied they were with the recording. Participants were allowed to rerecord their vocal imitations unlimited times before proceeding. Discarded imitations were saved as "drafts" on our server.

The second task addressed the use case where a user seeks to reproduce the exact sound of a specific instance of an audio concept (e.g. the sound of specific church bells). This task was similar to the first task, but instead of imitating the imagined sound of a description, participants were asked to listen to a reference *sound recording* (e.g. a recording of church bells) and to imitate it with their voice as closely as possible. They were then required to listen to both the reference recording and their own recorded imitation and indicate their satisfaction with the imitation. They were allowed to rerecord their vocal imitations unlimited times until satisfied. They were then asked to "describe the identity of the source of the reference audio" (using less than 5 words) and to indicate their confidence in their description on a discrete scale.

Each AMT Human Intelligence Task (HIT) consisted of at least four repetitions of the first task or four repetitions of the second task. If they hadn't completed one of these HITs in the last 24 hours, they also had an additional initial repetition of the task for practice to get them comfortable imitating sounds. Lastly, all participants completed a short survey the first time they performed one of these HITs. This survey contained questions regarding their age, gender, and experience with music technology, making music, and singing. Participants were allowed to contribute a maximum of 27 of each task (220 total imitations), but they were not allowed to contribute two of the same audio concept for a particular task.

At the beginning of each task we also asked participants to "find a quiet place with no background noise". Despite this, we did not have control of the participants recording environment. Therefore, while some background noise (recording hiss, etc.) is to be expected, we eliminated excessive background noise by requiring the signal plus noise to noise ratio (SNNR) of the recording to be at least 30dB. We also detected some instances of clipping in which the recording was distorted. If a recording was detected as clipped or did not meet the SNNR requirement, the participant was informed immediately, given some tips for improving their recording, and asked to rerecord their imitation.

5.4.1. Data Overview

Including all draft imitations, there were 10750 vocal imitations recorded by 248 unique participants. All of the submitted imitations (i.e. non-"drafts") were listened to by one of the authors, and any recordings lacking a vocal imitation or of poor recording quality were removed. The remaining recordings form the subset discussed in the remainder of the paper. This subset contains 4429 vocal imitations (2418 from the *sound recording* stimulus task, 2011 from the *sound label* stimulus task) recorded by 185 unique participants. Of the 175 participants that completed the survey, 100 identified as male / 75 as female, and their mean reported age was 31.8 (SD=8.5). The median number of imitations per participant was 4 (min=1, max=204). There were at least 10 (max of 11) vocal imitations collected for each of the 240 sound recordings and 200 sound labels.

5.5. Human Recognition of Vocal Imitations

The vocal imitations of audio concepts are just one half of communication path, the sender. To evaluate how well audio concepts can be communicated with vocal imitation, we also have to consider the receiver. While eventually we would like to build systems in which the receiver is a computer, in this work we instead use humans as the receiver, in order to establish a baseline level of communication effectiveness.

To establish this baseline, we had AMT workers perform several identification tasks.

For both the 2418 vocal imitations produced in response to reference sound recordings (a recording of a jackhammer) and the 2011 vocal imitations produced in response to descriptive labels (the word “jackhammer”), participants were presented a randomly selected vocal imitation. They could play this imitation unlimited times. They were asked to describe what it was an imitation of, using five words or less. They were then asked to indicate their degree of confidence in this free-response description on an 5-level scale.

Then, if the vocal imitation had been produced in response to a sound recording, the participant was presented 10 recordings drawn from the same audio concept subset (e.g. if it was an *everyday* sound they were presented 10 distinct everyday sounds): the referent recording and 9 random distractors. After hearing each recording in its entirety at least once, they were presented a 10-way forced choice to identify the recording the imitation was based on. Lastly, they had to indicate their confidence in their choice on a 5-level scale. Participants could play all of the recordings unlimited times.

Similarly, if the vocal imitation had been produced in response to a sound label, the participant was presented a 10-way forced choice between labels from the same audio concept subset. The task was to choose the label that the vocal imitation had been produced in response to. Participants could play the imitation recording unlimited times.

5.5.1. Data Overview

There were at least two (maximum of three) identification task instances assigned for each vocal imitation we collected. There were a total of 9174 identifications by 329 unique participants. The median number of identifications per participant was 10 (min=1, max=424).

5.6. Public Dataset

The dataset described in this chapter can be downloaded at `http://dx.doi.org/10.5281/zenodo.13862` and includes:

- Final and draft vocal imitations
- Descriptions of the referent sound recordings (with confidence ratings)

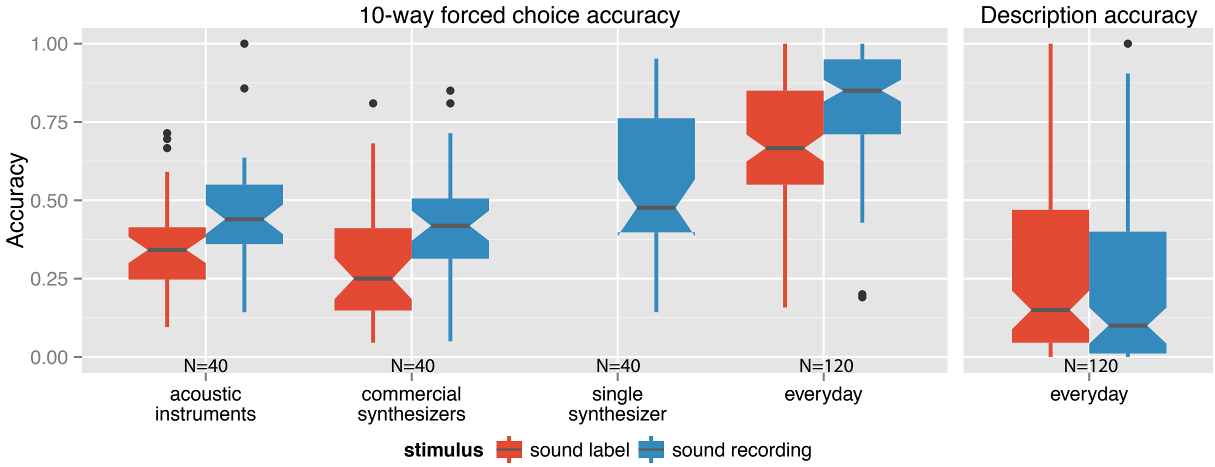


Figure 5.1. Human recognition accuracy of vocal imitations. The boxes extend from the lower to upper quartiles of the data. The dark grey lines in each box are the medians, and the notches around the median are 95% confidence intervals.

- Descriptions of the vocal imitations (with confidence ratings)
- 10-way forced choice identifications of the vocal imitations
- Participant background survey responses

5.7. Results

In this section, we provide a brief analysis of the human baseline performance on the identification of vocal imitations.

The authors of the *everyday* subset [**110**] published guidelines for scoring short descriptions of their sound recordings for binary recognition (e.g. description must contain “both the object (‘door’) and the closing action” for “door closing”). Using these guidelines, we scored our participants’ descriptions of the everyday sounds for recognition. We found that our participants’ mean recognition accuracy across the 120 *everyday* sounds was 0.80

(SD=0.25). This was similar to that of the previous study: mean=0.78 (SD=0.25). Comparing these to the previous study's results on the same 120 sounds, we obtained a Pearson correlation of r=0.84 (p=0.0), and a paired t-test of t(119)=1.67 (p=0.097). Therefore, our participants via AMT performed comparably to lab-consented participants, giving validity to the effort our participants put into the task.

Figure 5.1 shows the recognition accuracy for the vocal imitations, grouped by audio concept subset and stimulus type. "10-way forced-choice accuracy" refers to the recognition accuracy of the participants' 10-way forced-choice response in the identification task. For the *sound recording* stimulus vocal imitations, the mean recognition accuracy, broken down by audio concept subset was *acoustic instruments*: 0.45 (SD=0.18), *commercial synthesizers*: 0.42 (SD=0.18), *single synthesizer*: 0.54 (SD=0.22), *everyday*: 0.80 (SD=0.17), and mean accuracy for the *sound label* stimulus vocal imitations was *acoustic instruments*: 0.35 (SD=0.16), *commercial synthesizers*: 0.29 (SD=0.19), *everyday*: 0.68 (SD=0.20). Note that chance performance on all these tasks is 0.1.

Comparing each subset by stimulus type, the mean accuracy of the *sound recording* stimulus is greater than the *sound label* stimulus for all subsets (excluding *single synthesizer* since it lacks sound labels) according to one-sided paired t-tests, $p < 0.01$ in all cases. This difference is likely due participants' varied interpretations of what a text-based label means.

For the remainder of our analysis, we focus on the vocal imitations from the "sound recording" stimulus tasks. The *everyday* sounds were communicated the most effectively with vocal imitations. This may be due to the familiarity but also reproducibility of the *everyday* subset. Within that subset, imitations with the highest accuracy from the sound recording stimuli were typically sounds easily producible by the voice (human and animal sounds - e.g. “yawning”, “wolf”) or those with salient time-varied characteristics (e.g. “police siren”). Those with the lowest recognition accuracy were likely harder to accurately imitate with a single voice. For instance “glass breaking” (accuracy=0.20) has many overlapping small sonic events.

After a Welch’s f-test to test for equal means between the four audio concept subsets (p=0.0), we performed a pairwise t-test with Bonferroni correction and found that the difference of 0.12 in recognition rates between the *single synthesizer* (0.54) subset and the *commercial synthesizers* (0.42) subset bordered on statistical significance (p=0.058). Without an in depth acoustic analysis it is hard to establish why, but the data set is available to allow this follow-on work. One hypothesis is that the audio concepts in the *single synthesizer* class typically have simpler but strong modulation characteristics (i.e. salient temporal properties) which may have aided in the imitation and therefore recognition of these sounds. Several of the audio concepts that were difficult to recognize in the “single synthesizer” subset could be characterized as having complex timbres that didn’t change much over the course of a note.

In Figure 5.1, “Description accuracy” refers to the recognition accuracy of the participants’ free-response descriptions *of the vocal imitations* in the identification task. We again used the same scoring guidelines described in [**110**], and therefore we only scored the *everyday* subset, which was the same set used in their work. The mean recognition accuracy was 0.23 (SD=0.27) for the *sound recording* stimulus vocal imitations, and mean=0.27 (SD=0.27) for the *sound label* stimulus vocal imitations. Some audio concepts had a 0% recognition (e.g. “blinds”), while some had a 100% recognition (e.g. “sheep”).

While the free-response recognition accuracy is much lower than the forced-choice recognition accuracy, this is to be expected since the participants must describe the imitation without any additional clues. However, when failing to identify the referent audio concept, participants often described similar (e.g. "motorcycle" instead of "lawnmower") or possibly more general (e.g. "horn" instead of "boat horn") concepts. This implies that more information may be needed help to disambiguate or refine certain concepts. In an audio application, this could be achieved by asking the user for additional information.

5.8. Discussion

As this work is primarily about providing a data set for the community to use, the analysis in this work is intended to be illustrative rather than comprehensive. From our analysis, it seems *everyday* audio concepts were communicated the most effectively, though in a real application additional information may need to be provided by a user to disambiguate certain concepts. In the remaining instrumental subsets, audio concepts from the *single synthesizer* subset were communicated the most effectively, but further acoustic analysis is required to determine what enables some audio concepts to be communicated more effectively than others with vocal imitation.

5.9. Post-Research Related Work

Since my first publication on vocal imitation [**20**], the general topic of vocal imitation has gained significant interest that has resulted in several publications by other researchers, many of which have been influenced by SynthAssist and VocalSketch [**101, 139, 51, 204, 205, 111, 151, 112**]. Many of these publications are part of the SkatVG

[**172, 150**] project led by Davide Rocchesso which involves researchers in several institutions across Europe. The overall goal of the SkatVG project is to build computational support tools for sonic interaction design [**152**] in which users sketch sound ideas using both vocal imitation and gesture [**150**]. They have investigated vocal imitation from a number of different angles that have ranged from observing auditory sketching in sonic interaction design to investigating how we imitate sounds to building physical tools to support vocal imitation and gesture.

Delle Monache et al [**35, 122**] ran a workshop in which they first educated and familiarized participants with vocal imitation techniques and then asked the participants to design sonic interactions and observed the process [**35, 122**]. They found that participants benefited from the vocal-imitation familiarization training, but they also found that participants' use of gesture was limited.

However, researchers did investigate the combination of vocal imitation and gesture in more detail. Scurto [**165**] analyzed a large database of vocal and gestural imitations and found that voice was more precise than air gestures for communicating rhythmic information. They also found that participants would gesture in order to add additional descriptive information—e.g., "large" (opening hands wide) or "stronger" (a closed fist). Lastly, they also found that participants will *sometimes* split a layered sound into both a simultaneous vocal imitation and a gesture. From their analysis, vocal imitation seemed to be the dominant communication strategy, but they unfortunately did not quantify how gesture aids in recognition of audio concepts. Francoise [**51**] also investigated the combination of vocal imitation but his focus was the joint performance of vocal imitation and gesture and using gesture to resynthesize a previously performed vocal imitation.

Lastly, Rocchesso et al [**151**] identified four stages (*select*, *mimic*, *explore*, *refine*) in the sound design process during workshop and body-storming sessions, and they then developed *miMic*, an augmented microphone and system to support vocalization and gesticulation in their four stages of sound design.

Lemaitre et al [**101**], who are also part of the SkatVG project, have also continued their research. In a study published in 2016, they investigated the vocal imitation of specific auditory features [**101**]. They found that people do not simply mimic sounds, but instead “seek to emphasize the characteristic features of the referent sounds within the constraints of human vocal production”. For example, their study participants faithfully imitated the *pitch* and *tempo* of referent sounds, but transposed, compressed, or expanded *sharpness* into the participants’ register. In addition, participants poorly replicated the exact length of sound onsets; participants determined that the salient feature of the onset was whether it was categorized as short or long—the exact length didn’t matter. They stated that these results predict that vocal imitations of stationary sounds without any temporal evolution would be very difficult to recognize, and that systems that use vocalizations as an input cannot simply rely on the absolute values of the vocal imitation features. The implication of these results gives further validity to our assumptions that we used when designing SynthAssist.

Mauro and Rocchesso [**112**] presented a study on the clustering of a set of short (500ms) vocal imitations from a professional voice artist. Their eventual goal is to build a two-dimensional map of sounds that can also be indexed by vocal imitation. In this work though, they performed an automatic clustering using the median and interquartile range of 18 audio features and then compared the resulting clustering to a human-generated clustering. They found that humans had fair-to-moderate agreement (0.43 mean Cohen's Kappa) with each other, but just fair agreement (0.26 mean Cohen's Kappa) with the automatic clustering. However, these results provide a possible bound to the performance that a system can be expected to achieve given the limitations of human performance on this task.

Marchetto and Peeters [**111**] further investigated which audio features to use when representing vocal imitations. They first present a dataset of vocal imitations for 6 different synthesized sound classes (*up*, *down*, *up/down*, *impulse*, *repetition*, and *stable*), and asked 50 participants to vocally imitate each sound. They then presented methods for the classification / recognition of these imitations into the 6 classes. Using six different pitch and spectral features, they compared three different methods that used these features. They first computed local trend values (slopes from regression of 10 neighboring time series values) and then classified with an HMM. The second used global trends (slope values calculated over active regions which also use a splitting criteria) with an HMM. And the last method calculates "morphological" features from the time series. The time series features performed much better than the others and can be thought of as capturing the overall simple shapes of the different classes rather than the complexity of the evolution presented using the HMM.

Lastly, researchers have already begun using the VocalSketch dataset to further research in this area. Zhang and Duan used the dataset to build a sound retrieval system in which audio concepts are queried using vocal imitation. Using the VocalSketch dataset, they trained a deep neural networks (DNNs) (stacked auto-encoder (SAE)) to automatically learn features and then used supervised (support vector machine (SVM)) [**204**] and unsupervised (Kullback-Leibler divergence (KL divergence) and dynamic time warping (DTW) distance) sound classification and retrieval approaches [**205**] to return the audio concepts of the vocal imitations input to the system. The two systems obtained comparable results to each other and both performed better than a baseline system that used mel-frequency cepstral coefficient (MFCC) features.

5.10. Conclusions

In this chapter, I presented VocalSketch, a novel data set containing thousands of crowdsourced vocal imitations of audio concepts and identifications of those imitations by other humans—the first comprehensive and largest publicly available dataset of vocal imitations. This dataset provides the research community with a baseline for query-by-vocal-imitation audio retrieval systems and a labeled dataset for supervised machine learning. This data could be used to learn more robust mappings from vocal imitations to referent audio and to synthesis parameters, allowing us to search the continuous parameter space of synthesizers rather than the sampled discretized space used in SynthAssist. But more generally, VocalSketch is a necessary and important step towards understanding which audio concepts can be effectively communicated with vocal imitation and what the characteristics of these audio concepts are, and it will inform us as to which applications can take advantage of this intuitive form of "auditory sketching".

CHAPTER 6

Conclusions

My research goal is for creative media production tools to aid in creativity rather than impede it. When tools require users to communicate their goals using difficult-to-understand parameters, they frustrate novice users and impede creativity. In this dissertation, I have addressed the problem of audio production tools' lack of affordances for novices. **I have proposed to solve this problem by enabling audio production tools to understand the same methods of communication that novices use when communicating sounds to other people: *descriptive language*, *evaluative feedback*, and examples such as *vocal imitation*.** This is a fundamental rethinking of the audio production interaction paradigm. This approach allows tool novices[1] to communicate their sound ideas to software in familiar ways rather than using low-level technical parameters that require years of experience to use effectively.

However, the work in this dissertation does have its limitations. For example, one limitation of this work is that it focuses on communication for "low threshold, high ceiling" interfaces but neglects the other recommended design criteria of creativity support tools, e.g. *support exploratory search* [**170, 147**]. Decomposing the creative process into parts and focusing on communications allowed me to easily test that one component of the problem. But communication is only step in the process. In fact, in the ideal scenario

[1] As noted in Section 1.2.2, I define *tool novices* as users of audio production tools that lack the experience and knowledge of audio production tools. This is in contrast to *domain novices* that I defined as users that lack the experience and knowledge of musical practice.

in which the user is able to perfectly communicate a desired audio concept to a limitless audio processor, communication alone would only be sufficient if the user knew exactly what they desired. However, users may not know exactly what they want. Like any design task, creative audio production may be ill-defined (i.e. both the problem and the solution are not known at the outset). Design thinking processes of ideation (exploration) and refinement can help overcome this kind of problem [**30**], and supporting such processes would also address the first design principle of creativity support tools: *support exploratory search* [**170**].

SocialEQ already supports exploration. Audio descriptors can be used to query equalization settings directly, which allows for different options to be explored quickly. In addition, I also built a two-dimensional map which can be utilized for exploring the space of audio equalization concepts.

Supporting exploration in SynthAssist is more difficult. Since multiple referent sounds may map onto the same vocal imitation, querying with vocal imitation requires additional information to communicate an audio concept. In SynthAssist, users provide this additional information through feedback to the system. However, with SynthAssist's current retrieval algorithm, it may take several iterations of rating to retrieve a specific sound. Therefore, *quickly* trying out different imitations to explore different settings isn't feasible. A solution to this could simply be to make a better retrieval algorithm that doesn't require as much feedback. With the data from VocalSketch, it may be possible to learn mappings from vocal imitations to the referent audio. By training a system with this data, we can possibly reduce the amount of feedback that the system needs, and we may also be able to search the full parameter space rather than the sampled parameter space (another limitation of SynthAssist).

Another possible solution to supporting exploration in SynthAssist is to use the current SynthAssist algorithm in tandem with alternative parameterized interfaces such as two-dimensional maps or perceptual / semantic controllers to control the synthesizer. The current SynthAssist algorithm could be used to move to a particular region of the synthesis space, and the parameterized controllers can be used to meaningfully explore the region.

Lastly, a user can also guide the current SynthAssist algorithm based on their feedback ratings in order to explore the synthesis space. However, this can create a moving target which can confuse the retrieval algorithm. I believe this "problem" of *concept evolution* [**98**] exists in many creative evaluation-based interfaces, and there is anecdotal evidence from my user studies that supports this hypothesis [**23**]. It is common for the teacher's understanding of a concept to evolve and change as they teach. This is especially common for creative tasks—preferred goals and methods can and should shift during the creative process. However, this can be problematic for interactively training a machine learning system to assist a creative task. Algorithms typically presume a constant goal and treat inconsistency in training data as unwanted noise. "Creative types" typically don't understand the internals of learning algorithms and cannot compensate for the weakness of the algorithms. Therefore, we must develop methods better able to handle training data that represents a shifting goal or concept. Ideally, these approaches should incorporate a training paradigm that even novice, non-technical users can use effectively.

Moving forward, I plan to address the limitations highlighted above and focus on *exploration*, and *refinement* in audio production interfaces in addition to *communication* in my future research. When doing so, I also plan to evaluate these new interfaces with users in realistic creative contexts rather than by the evaluation-by-parts approach I have taken thus far.

In addition to supporting novices, I also plan to broaden participation in the production of creative audio content by other populations whose needs have not been adequately met by commercial software companies (e.g., users with visual impairments and users with motor impairments). This dissertation has laid the groundwork for audio production interfaces with which tool novices can communicate more easily, but much more research is needed to ensure that all aspects of the audio production process are accessible to all individuals. While novice users lack experience and knowledge to use standard audio production interfaces, other users may lack other traits/abilities required to use standard audio production tools. For example, visually impaired users may find it difficult to control audio production tools such as effects and synthesizers that typically have many tunable parameters. While such a user may understand signal processing theory, the graphical user interfaces of audio plugins are often not accessible to screen readers, and even when they are, the large number of idiosyncratic parameters can be difficult to navigate. Similar to my SynthAssist work with novices, one solution to this problem would be to allow users to communicate their synthesis ideas with vocal imitation and evaluative feedback. However, this approach would clearly not work for other physically impaired users. For example, users with advanced ALS (a progressive neurodegenerative disease) usually have fine eyesight but are slow to control graphical interfaces with many parameters (using eye-tracking for example) and would be unable to imitate their desired sound with their voice. For these users, a set vocabulary of audio descriptors, coarse two-dimensional maps, haptic interfaces, or simple pairwise comparisons for feedback/refinement may be more appropriate for communicating their ideas. By researching effective interaction methods for these populations, I hope to enable these often overlooked populations to more easily creatively express themselves through audio.

Lastly, many of methods I have described in this dissertation are not limited to audio production and may be applicable to other domains of creative media production (e.g., graphics, image, and video) in which users must express their creative goals in the form of low-level parameters. However, I have not yet adapted and tested these methods on these other domains. In future work, I plan to address this knowledge gap and learn the strengths and weaknesses of these techniques in these domains. Moving forward, I seek to investigate methods, interactions, and design principles that are general to all types of creative media production.

In closing, as technology has progressed, our creative media production tools have become more powerful and more economically affordable. However, for many potential users, these tools are inaccessibly complex. Simpler tools exist, but they achieve their simplicity through limited and prescriptive functionality, impeding creativity. I seek a future in which these tools don't impede our creativity, but rather aid it—they collaborate with us, understand our sometimes vague goals, and even help us further define and achieve our goals. There are enough forces in people's lives that impede creativity, our tools should not be one of them.

References

[1] Ableton. Ableton, 2013. http://www.ableton.com.

[2] American National Standard / Acoustical Society of America. Acoustical terminology, 2013.

[3] Anonymous, 2014.

[4] Appelbaum, M. Pre-composition, 2003.

[5] Apple, Inc. Garageband 10.1.0, 2015.

[6] Barchiesi, D., and Reiss, J. Automatic target mixing using least-squares optimization of gains and equalization settings. *Proceedings of the Conference on Digital Audio Effects* (2009), 7–14.

[7] Bell, A. P. Can we afford these affordances? garageband and the double-edged sword of the digital audio workstation. *Action, Criticism, and Theory for Music Education 14*, 1 (2015), 44.

[8] Bell, A. P., Hein, E., and Ratcliffe, J. Beyond skeuomorphism: The evolution of music production software user interface metaphors. *Journal on the Art of Record Production 9* (2015).

[9] Bencina, R. The metasurface: applying natural neighbour interpolation to two-to-many mapping. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (2005).

[10] Birmingham, W., Dannenberg, R., and Pardo, B. Query by humming with the vocalsearch system. *Communications of the ACM 49*, 8 (2006), 49–52.

[11] Bitzer, J., and LeBoeuf, J. Automatic detection of salient frequencies. In *Proceedings of the Audio Engineering Society Convention 126* (2009).

[12] Blancas, D. S., and Janer, J. Sound retrieval from voice imitation queries in collaborative databases. In *Proceedings of the International Conference of the Audio Engineering Society* (2014).

[13] Borg, I., and Groenen, P. *Modern multidimensional scaling: Theory and applications.* Springer Verlag, 2005.

[14] Bradley, R. A., and Terry, M. E. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika 39*, 3-4 (1952), 324–345.

[15] Burgess, R. J. *The history of music production.* Oxford University Press, New York, 2014.

[16] Buxton, W. *Sketching user experience : getting the design right and the right design.* Elsevier/Morgan Kaufmann, Amsterdam ; Boston, 2007.

[17] Cage, J. *The Future of Music: Credo*. Audio culture: Readings in modern music. Continuum, New York, 2004.

[18] Cartwright, M., and Pardo, B. Social-EQ: Crowdsourcing an equalization descriptor map. In *Proceedings of the International Society for Music Information Retrieval Conference* (2013).

[19] Cartwright, M., and Pardo, B. Synthassist: an audio synthesizer programmed with vocal imitation. In *Proceedings of the ACM International Conference on Multimedia* (2014), ACM, pp. 741–742.

[20] Cartwright, M., and Pardo, B. Synthassist: Querying an audio synthesizer by vocal imitation. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (London, 2014).

[21] Cartwright, M., and Pardo, B. Translating sound adjectives by collectively teaching abstract representations. In *Proceedings of the Collective Intelligence Conference* (2014).

[22] Cartwright, M., and Pardo, B. VocalSketch: Vocally imitating audio concepts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 43–46.

[23] Cartwright, M., and Pardo, B. The moving target in creative interactive machine learning. In *Proceedings of the Workshop on Human-Centred Machine Learning at CHI 2016* (2016).

[24] Cartwright, M., Pardo, B., Mysore, G., and Hoffman, M. Fast and easy crowdsourced perceptual audio evaluation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2016).

[25] Cartwright, M. B., and Pardo, B. A. Systems, methods, and apparatus to search audio synthesizers using vocal imitation, 2016.

[26] Casey, M. A. Sound classification and similarity. In *Introduction to MPEG-7: Multimedia Content Description*, B. S. Manjunath, P. Salembier, and T. Sikora, Eds. Wiley, West Sussex, England, 2002.

[27] Chaudhuri, S., Kalogerakis, E., Giguere, S., and Funkhouser, T. Attribit: content creation with semantic attributes. In *Proceedings of the ACM symposium on User interface software and technology* (2013), ACM, pp. 193–202.

[28] Chen, K.-T., Wu, C.-C., Chang, Y.-C., and Lei, C.-L. A crowdsourceable qoe evaluation framework for multimedia content. In *Proceedings of the ACM international conference on Multimedia* (2009), ACM.

[29] Chen, X., Bennett, P. N., Collins-Thompson, K., and Horvitz, E. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the ACM international conference on Web search and data mining* (2013), ACM.

[30] Cross, N. *Design thinking : understanding how designers think and work*. Berg, Oxford; New York, 2011.

[31] Dahlstedt, P. Evolution in creative sound design. *Evolutionary computer music* (2007), 79–99.

[32] David, H. A. *The method of paired comparisons*, vol. 12. DTIC Document, 1963.

[33] Davis, N., Winnemller, H., Dontcheva, M., and Do, E. Y.-L. Toward a cognitive theory of creativity support. In *Proceedings of the ACM Conference on Creativity & Cognition* (2013), ACM, pp. 13–22.

[34] Dawid, A. P., and Skene, A. M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*, 1 (1979), 20–28.

[35] Delle Monache, S., Baldan, S., Mauro, D. A., and Rocchesso, D. A design exploration on the effectiveness of vocal imitations. In *Proceedings of the International Conference on Computer Music, and Sound and Music Computing Conference* (2014), Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

[36] Dessein, A., and Lemaitre, G. Free classification of vocal imitations of everyday sounds. In *Proceedings of the Sound and Music Computing Conference* (2009).

[37] developers, P. Pystan: the python interface to stan, version 2.6.3, 2015.

[38] Disley, A., and Howard, D. Spectral correlates of timbral semantics relating to the pipe organ. *Speech, Music and Hearing 46* (2004), 25–39.

[39] Eigenfeldt, A., and Pasquier, P. Real-time timbral organisation: Selecting samples based upon similarity. *Organised Sound 15*, 2 (2010), 159–166.

[40] Eitz, M., Hays, J., and Alexa, M. How do humans sketch objects? *ACM Transaction on Graphics 31*, 4 (2012), 44.

[41] Ekman, I., and Rinott, M. Using vocal sketching for designing sonic interactions. In *Proceedings of the ACM Conference on Designing Interactive Systems* (2010), pp. 123–131.

[42] Emiya, V., Vincent, E., Harlander, N., and Hohmann, V. Subjective and objective quality assessment of audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing 19*, 7 (2011), 2046–2057.

[43] Eno, B. *The studio as compositional tool.* Audio culture: Readings in modern music. Continuum, New York, 2004.

[44] Esling, P., and Agon, C. Multiobjective time series matching for audio classification and retrieval. *IEEE Transactions on Speech Audio and Language Processing 21*, 10 (2013), 2057–2072.

[45] Ethington, R., and Punch, B. Seawave: A system for musical timbre description. *Computer Music Journal 18*, 1 (1994), 30–39.

[46] Fasciani, S., and Wyse, L. A voice interface for sound generators: adaptive and automatic mapping of gestures to sound. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (2012).

[47] Fiebrink, R. A. *Real-time human interaction with supervised learning algorithms for music composition and performance.* PhD thesis, Princeton Univ., 2011.

[48] Figuerola Salas, O., Adzic, V., and Kalva, H. Subjective quality evaluations using crowdsourcing. In *Proceedings of the Picture Coding Symposium* (2013).

[49] Fogarty, J., Tan, D., Kapoor, A., and Winder, S. CueFlik: interactive concept learning in image search. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2008), ACM, pp. 29–38.

[50] Fox, B., Sabin, A., Pardo, B., and Zopf, A. *Modeling perceptual similarity of audio signals for blind source separation evaluation.* Springer, 2007, pp. 454–461.

[51] Francoise, J. *Motion-Sound Mapping by Demonstration.* Thesis, Universit Pierre Et Marie Curie, 2015.

[52] Fritts, L. University of iowa musical instrument samples, 2012. `http://theremin.music.uiowa.edu/MIS.html`.

[53] Garcia, R. Growing sound synthesizers using evolutionary methods. In *Proceedings of the Workshop on Artificial Life Models for Musical Applications* (2001).

[54] Gelman, A. *Bayesian data analysis*, 3rd ed. Chapman & Hall/CRC texts in statistical science. CRC Press, Boca Raton, 2014.

[55] Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. Query by humming: musical information retrieval in an audio database. In *Proceedings of the ACM International Conference on Multimedia* (1995).

[56] Gibson, J. J. *The ecological approach to visual perception.* Houghton Mifflin, Boston, 1979.

[57] Gillet, O., and Richard, G. Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems 24*, 2-3 (2005), 159–177.

[58] Glasberg, B., and Moore, B. Derivation of auditory filter shapes from notched-noise data. *Hearing Research 47*, 12 (1990), 103–138.

[59] Google Inc. Google translate, 2014. `http://translate.google.com`.

[60] Gounaropoulos, A., and Johnson, C. Synthesising timbres and timbre-changes from adjectives/adverbs. In *Proceedings of the Workshop on Applications of Evolutionary Computation* (2006), Springer, pp. 664–675.

[61] Grey, J. M. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America 61*, 5 (1977), 1270–1277.

[62] Hafezi, S., and Reiss, J. D. Autonomous multitrack equalization based on masking reduction. *Journal of the Audio Engineering Society 63*, 5 (2015), 312–323.

[63] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning.* Springer New York, 2001.

[64] Hayes, A. F., and Krippendorff, K. Answering the call for a standard reliability measure for coding data. *Communication methods and measures 1*, 1 (2007), 77–89.

[65] Heise, S., Hlatky, M., and Loviscach, J. Aurally and visually enhanced audio search with soundtorch. In *Extended Abstracts on Human factors in Computing Systems* (2009).

[66] Heise, S., Hlatky, M., and Loviscach, J. Automatic cloning of recorded sounds by software synthesizers. In *Proceedings of the Audio Engineering Society Convention 127* (2009).

[67] Helén, M., and Lahti, T. Query by example methods for audio signals. In *Proceedings of the Nordic Signal Processing Symposium* (2006).

[68] Helén, M., and Virtanen, T. Query by example of audio signals using euclidean distance between gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2007).

[69] Helén, M., and Virtanen, T. A similarity measure for audio query by example based on perceptual coding and compression. In *Proceedings of the Conference on Digital Audio Effects* (2007).

[70] Helmholtz, H., and Ellis, A. *On the sensations of tone as a physiological basis for the theory of music*, 2nd english ed. Dover, New York, 1954.

[71] Herrera-Boyer, P., Peeters, G., and Dubnov, S. Automatic classification of musical instrument sounds. *Journal of New Music Research 32*, 1 (2003), 3 – 21.

[72] Hershey, J., and Olsen, P. Approximating the kullback leibler divergence between gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (2007).

[73] Hoffman, M. D., and Gelman, A. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Resesearch 15*, 1 (2014), 1593–1623.

[74] Horner, A., Beauchamp, J., and Haken, L. Machine tongues xvi: Genetic algorithms and their application to fm matching synthesis. *Computer Music Journal 17*, 4 (1993), 17–29.

[75] Hossfeld, T., Hirth, M., Korshunov, P., Hanhart, P., Gardlo, B., Keimel, C., and Timmerer, C. Survey of web-based crowdsourcing frameworks for subjective quality assessment. In *Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop on* (2014).

[76] Hofeld, T., Keimel, C., Hirth, M., Gardlo, B., Habigt, J., Diepold, K., and Tran-Gia, P. Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing. *IEEE Transactions on Multimedia 16*, 2 (2014), 541–558.

[77] Huang, C.-Z. A., Duvenaud, D., Arnold, K. C., Partridge, B., Oberholtzer, J. W., and Gajos, K. Z. Active learning of intuitive control knobs for

synthesizers using gaussian processes. In *Proceedings of the International Conference on Intelligent User Interfaces* (Haifa, Israel, 2014).

[78] Huber, D., and Runstein, R. *Modern recording techniques*, 7th ed. Focal Press/Elsevier, Amsterdam ; Boston, 2010.

[79] Huq, A., Cartwright, M., and Pardo, B. Crowdsourcing a real-world online query by humming system. In *Proceedings of the Sound and Music Computing Conference* (Barcelona, 2010).

[80] ITU. Recommendation ITU-R BS.1534-1: Method for the subjective assessment of intermediate quality level of coding systems, 2003.

[81] ITU. Recommendation ITU-R BS.1116-2: Methods for the subjective assessment of small impairments in audio systems, 2014.

[82] ITU. Recommendation ITU-R BS.1534-2: Method for the subjective assessment of intermediate quality level of audio systems, 2014.

[83] Izhaki, R. *Mixing audio : concepts, practices and tools*, 2nd ed. Focal Press, Amsterdam ; Boston, 2008. 2011933235 GBB177522 Roey Izhaki. ill. ; 25 cm. Previous ed.: 2008. Includes index.

[84] Jackson, M. Michael jackson and the dangerous court case deposition 1994 - part 1/5.

[85] Janer Mestres, J. *Singing-driven interfaces for sound synthesizers.* PhD thesis, Universitat Pompeu Fabra, 2008.

[86] Jehan, T., and Schoner, B. An audio-driven perceptually meaningful timbre synthesizer. In *Proceedings of the International Computer Music Conference* (2001).

[87] Jillings, N., De Man, B., Moffat, D., and Reiss, J. D. Web audio evaluation tool: A browser-based listening test environment. In *Proceedings of the Sound and Music Computing Conference* (2015).

[88] Johnson, C. G., and Gounaropoulos, A. Timbre interfaces using adjectives and adverbs. In *Proceedings of the International Confernces on New Interfaces for Musical Expression* (2006).

[89] Johnson, G., Gross, M. D., Hong, J., and Yi-Luen Do, E. Computational support for sketching in design: a review. *Foundations and Trends in Human-Computer Interaction 2*, 1 (2009), 1–93.

[90] Jones, S. *Rock formation : music, technology, and mass communication.* Foundations of popular culture. Sage, Newbury Park, Calif., 1992.

[91] Kealy, E. R. *The real rock revolution: sound mixers, social inequality, and the aesthetics of popular music production.* Thesis, Northwestern University, 1974.

[92] Keimel, C., Habigt, J., Horch, C., and Diepold, K. Qualitycrowd: A framework for crowd-based quality evaluation. In *Proceedings of the Picture Coding Symposium* (2012).

[93] Kim, B., and Pardo, B. Speeding learning of personalized audio equalization. In *Proceedings of the IEEE International Conference on Machine Learning and Applications* (2014), pp. 495–499.

[94] Kim, H.-G., Moreau, N., and Sikora, T. *MPEG-7 audio and beyond : audio content indexing and retrieval.* J. Wiley, Chichester, West Sussex, England ; Hoboken, NJ, USA, 2005.

[95] Kraft, S., and Zlzer, U. Beaqlejs: Html5 and javascript based framework for the subjective evaluation of audio quality. In *Linux Audio Conference, Karlsruhe, DE* (2014).

[96] Krippendorff, K. *Content analysis : an introduction to its methodology*, 3rd ed. SAGE, Los Angeles ; London, 2013.

[97] Krumhansl, C. L. Why is musical timbre so hard to understand? In *Structure and Perception of Electroacoustic Sound and Music*, S. Nielzen and O. Olsson, Eds. Elsevier, Amsterdam, 1989, pp. 43–53.

[98] Kulesza, T., Amershi, S., Caruana, R., Fisher, D., and Charles, D. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the ACM conference on Human factors in computing systems* (2014), ACM, pp. 3075–3084.

[99] Laffont, P.-Y., Ren, Z., Tao, X., Qian, C., and Hays, J. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics 33*, 4 (2014), 149.

[100] Lemaitre, G., Dessein, A., Susini, P., and Aura, K. Vocal imitations and the identification of sound events. *Ecological Psychology 23*, 4 (2011), 267–307.

[101] Lemaitre, G., Jabbari, A., Misdariis, N., Houix, O., and Susini, P. Vocal imitations of basic auditory features. *The Journal of the Acoustical Society of America 139*, 1 (2016), 290–300.

[102] Lemaitre, G., and Rocchesso, D. On the effectiveness of vocal imitations and verbal descriptions of sounds. *The Journal of the Acoustical Society of America 135*, 2 (2014), 862–873.

[103] Lemaitre, G., Susini, P., Rocchesso, D., Lambourg, C., and Boussard, P. Using vocal imitations for sound design. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research* (2013).

[104] Lemaitre, G., Susini, P., Rocchesso, D., Lambourg, C., and Boussard, P. Non-verbal imitations as a sketching tool for sound design. In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval* (2014), Springer International Publishing, pp. 558–574.

[105] Letowski, T., and Mikiewicz, A. Timbre solfege: A course in perceptual analysis of sound. *Signal Processing in Sound Engineering, J. Adamczyk (Ed.). Institute of*

Fundamental Technological Research, Polish Academy of Sciences, Warsaw, Poland (2013).

[106] Levy, S. Push-button rock. *Rolling Stone* (November 21 1985).

[107] Lukasik, E. Towards timbre-driven semantic retrieval of violins. In *Proceedings of the International Conference on Intelligent Systems Design and Applications* (2005).

[108] Ma, Z., Reiss, J. D., and Black, D. A. Implementation of an intelligent equalization tool using yule-walker for music mixing and mastering. In *Proceedings of the Audio Engineering Society Convention 134* (2013), Audio Engineering Society.

[109] Manning, C., Raghavan, P., and Schtze, H. *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

[110] Marcell, M. M., Borella, D., Greene, M., Kerr, E., and Rogers, S. Confrontation naming of environmental sounds. *Journal of Clinical and Experimental Neuropsychology 22*, 6 (2000), 830–864.

[111] Marchetto, E., and Peeters, G. A set of audio features for the morphological description of vocal imitations. In *Proceedings of the International Conference on Digital Audio Effects* (2015).

[112] Mauro, D., and Rocchesso, D. Analyzing and organizing the sonic space of vocal imitations. In *Proceedings of the Audio Mostly 2015 on Interaction With Sound* (2015), ACM, p. 23.

[113] McAdams, S., Winsberg, S., Donnadieu, S., Soete, G., and Krimphoff, J. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research 58*, 3 (1995), 177–192.

[114] McDermott, J., O'Neill, M., and Griffith, N. Interactive ec control of synthesized timbre. *Evolutionary Computation 18*, 2 (2010), 277–303.

[115] McGrenere, J., and Ho, W. Affordances: Clarifying and evolving a concept. In *Graphics Interface* (2000), vol. 2000, pp. 179–186.

[116] McLeod, P., and Wyvill, G. A smarter way to find pitch. In *Proceedings of the International Computer Music Conference* (2005).

[117] Mecklenburg, S., and Loviscach, J. subjeqt: controlling an equalizer through subjective terms. In *Extended Abstracts on Human Factors in Computing Systems* (Montreal, Canada, 2006).

[118] Miller, G. Wordnet: a lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41.

[119] Mintz, D. Toward timbral synthesis: a new method for synthesizing sound based on timbre description schemes. Master's thesis, University of California, 2007.

[120] Momeni, A., and Wessel, D. Characterizing and controlling musical material intuitively with geometric models. In *Proceedings of the International Confernces on New Interfaces for Musical Expression* (2003).

[121] Monache, S. D., Polotti, P., and Rocchesso, D. A toolkit for explorations in sonic interaction design. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound* (2010), ACM, p. 1.

[122] Monache, S. D., Rocchesso, D., Baldan, S., and Mauro, D. Growing the practice of vocal sketching. In *International Conference on Auditory Display* (2015).

[123] Moore, B. C. *An introduction to the psychology of hearing*. Brill, 2012.

[124] Moorefield, V. *The producer as composer : shaping the sounds of popular music*, 1st mit press pbk. ed. MIT Press, Cambridge, Mass., 2010.

[125] Müller, M. Dynamic time warping. In *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007, pp. 69–84.

[126] Niennattrakul, V., and Ratanamahatana, C. A. Shape averaging under time warping. In *Proceedings of the International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* (2009).

[127] Norman, D. A. *The design of everyday things*. Basic books, 1988.

[128] O'Donovan, P., Lbeks, J., Agarwala, A., and Hertzmann, A. Exploratory font selection using crowdsourced attributes. *ACM Transactions on Graphics 33*, 4 (2014), 92.

[129] Osting, B., Brune, C., and Osher, S. J. Optimal data collection for informative rankings expose well-connected graphs. *Journal of Machine Learning Research 15*, 1 (2014), 2981–3012.

[130] Owsinski, B. *The Mixing Engineer's Handbook*. Mix Pro Audio Series. Mix Books, Vallejo, 1999.

[131] Paolacci, G., Chandler, J., and Ipeirotis, P. G. Running experiments on amazon mechanical turk. *Judgment and Decision making 5*, 5 (2010), 411–419.

[132] Pardo, B., and Birmingham, W. Query by humming: How good can it get? In *Proceedings of the Workshop on Music Information Retrieval* (2003), vol. 1001, Citeseer, p. 107.

[133] Pardo, B., Little, D., and Gergle, D. Building a personalized audio equalizer interface with transfer learning and active learning. In *Proceedings of the ACM workshop on Music information retrieval with user-centered and multimodal strategies* (2012), ACM, pp. 13–18.

[134] Pardo, B., Little, D., and Gergle, D. Towards speeding audio eq interface building with transfer learning. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (2012), vol. 10, p. 11.

[135] Pardo, B., Shifrin, J., and Birmingham, W. Name that tune: A pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology 55*, 4 (2004), 283–300.

[136] Passonneau, R. J., and Carpenter, B. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics 2* (2014), 311–326.

[137] Peeters, G. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM, 2003.

[138] Perez-Gonzalez, E., and Reiss, J. Automatic equalization of multichannel audio using cross-adaptive methods. In *Proceedings of the Convention of the Audio Engineering Society* (2009), Audio Engineering Society.

[139] Piccolo, A. D., and Rocchesso, D. Non-speech voice for sonic interaction: a catalogue. *Journal on Multimodal User Interfaces* (2016), 1–17.

[140] Porcello, T. Speaking of sound language and the professionalization of sound-recording engineers. *Social Studies of Science 34*, 5 (2004), 733–758.

[141] Press., O. U. Oxford english dictionary, 1992.

[142] Qi, H., Hartono, P., Suzuki, K., and Hashimoto, S. Sound database retrieved by sound. *Acoustical Science and Technology 23*, 6 (2002), 293–300.

[143] Qianqian, X., Qingming, H., Tingting, J., Bowei, Y., Weisi, L., and Yuan, Y. Hodgerank on random graphs for subjective video quality assessment. *Multimedia, IEEE Transactions on 14*, 3 (2012), 844–857.

[144] Rafii, Z., and Pardo, B. Learning to control a reverberator using subjective perceptual descriptors. In *Proceedings of the International Society for Music Information Retrieval Conference* (2009), pp. 285–290.

[145] Reed, D. Capturing perceptual expertise: a sound equalization expert system. *Knowledge-Based Systems 14*, 12 (2001), 111–118.

[146] Reinecke, K., and Gajos, K. Z. Labinthewild: Conducting large-scale online experiments with uncompensated samples. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing* (2015), ACM.

[147] Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., and Eisenberg, M. Design principles for tools to support creative thinking. *Report of Workshop on Creativity Support Tools* (2005).

[148] Ribeiro, F., Florencio, D., Cha, Z., and Seltzer, M. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2011), pp. 2416–2419.

[149] Roberts, F. S. *Measurement theory with applications to decisionmaking, utility, and the social sciences.* Encyclopedia of mathematics and its applications. Cambridge University Press, Cambridge Cambridgeshire ; New York, NY, USA, 1984.

[150] Rocchesso, D., Lemaitre, G., Susini, P., Ternstrm, S., and Boussard, P. Sketching sound with voice and gesture. *Interactions 22*, 1 (2015), 38–41.

[151] Rocchesso, D., Mauro, D. A., and Monache, S. D. mimic: The microphone as a pencil. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction* (2016), ACM, pp. 357–364.

[152] Rocchesso, D., Serafin, S., Behrendt, F., Bernardini, N., Bresin, R., Eckel, G., Franinovic, K., Hermann, T., Pauletto, S., and Susini, P. Sonic interaction design: sound, information and experience. In *Extended Abstracts on Human Factors in Computing Systems* (2008), ACM, pp. 3969–3972.

[153] Rotondi, J., Anjos, A. A., Thorngren, E., Killen, K., and Martine, T. How pros use reverb: Expert mixers reveal the power and perils of the magical effect, 2016.

[154] Rui, Y., and Huang, T. Optimizing learning in image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2000), vol. 1, pp. 236–243 vol.1.

[155] Russolo, L. *The Art of Noises: Futurist Manifesto.* Audio culture: Readings in modern music. Continuum, New York, 2004.

[156] Sabin, A., Rafii, Z., and Pardo, B. Weighting-function-based rapid mapping of descriptors to audio processing parameters. *Journal of the Audio Engineering Society 59*, 6 (2011), 419–430.

[157] Sabin, A. T., and Pardo, B. 2DEQ: an intuitive audio equalizer. In *Proceeding of the ACM conference on Creativity and cognition* (2009), ACM, pp. 435–436.

[158] Sabin, A. T., and Pardo, B. A method for rapid personalization of audio equalization parameters. In *Proceedings of the ACM International Conference on Multimedia* (2009).

[159] Salas, O. F., Adzic, V., Shah, A., and Kalva, H. Assessing internet video quality using crowdsourcing. In *Proceedings of the ACM international workshop on Crowdsourcing for multimedia* (2013), ACM.

[160] Sarkar, M., Vercoe, B., and Yang, Y. Words that describe timbre: a study of auditory perception through language. In *Proceedings of the Language and Music as Cognitive Systems Conference* (2007).

[161] Schoeffler, M., Stter, F.-R., Bayerlein, H., Edler, B., and Herre, J. An experiment about estimating the number of instruments in polyphonic music: A comparison between internet and laboratory results. In *Proceedings of the International Society for Music Information Retrieval Conference* (2013).

[162] Schoeffler, M., Stter, F.-R., Edler, B., and Herre, J. Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R Recommendation BS. 1534 (MUSHRA). In *Proceedings of the Web Audio Conference* (2015).

[163] Schwarz, D., and Schnell, N. Sound search by content-based navigation in large databases. In *Proceedings of the Sound and Music Computing Conference* (2009).

[164] Scott, J., Migneco, R., Morton, B., Hahn, C., Diefenbach, P., and Kim, Y. An audio processing library for MIR application development in flash. In *Proceedings of the International Society of Music Information Retrieval Conference* (2010).

[165] Scurto, H. Combination of gesture and vocalization in the imitation of sounds. Thesis, IRCAM, 2015.

[166] Seetharaman, P., and Pardo, B. Crowdsourcing a reverberation descriptor map. In *Proceedings of the ACM International Conference on Multimedia* (2014), ACM, pp. 587–596.

[167] Seetharaman, P., and Pardo, B. Audealize: Crowdsourced audio production tools. *Journal of the Audio Engineering Society* (2016).

[168] Shepard, R. Geometrical approximations to the structure of musical pitch. *Psychological Review 89*, 4 (1982), 305–333.

[169] Shneiderman, B. Creating creativity: user interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction 7*, 1 (2000), 114–138.

[170] Shneiderman, B. Creativity support tools: accelerating discovery and innovation. *Communications of the ACM 50*, 12 (2007), 20–32.

[171] Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., Edmonds, E., Eisenberg, M., Giaccardi, E., Hewett, T., Jennings, P., Kules, B., Nakakoji, K., Nunamaker, J., Pausch, R.,

Selker, T., Sylvan, E., and Terry, M. Creativity support tools: Report from a u.s. national science foundation sponsored workshop. *International Journal of Human-Computer Interaction 20*, 2 (2006), 61 – 77.

[172] SkatVG. SkatVG - sketching audio technologies using vocalizations and gestures, 2016.

[173] Smalley, D. Spectromorphology: explaining sound-shapes. *Organised Sound 2*, 02 (1997), 107–126.

[174] Smaragdis, P., and Mysore, G. J. Separation by "humming": User-guided sound extraction from monophonic mixtures. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2009), pp. 69–72.

[175] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), Association for Computational Linguistics.

[176] Solomon, L. Search for physical correlates to psychological dimensions of sounds. *The Journal of the ASA 31*, 4 (1959), 492–497.

[177] Stables, R., Enderby, S., De Man, B., Reiss, J., Fazekas, G., and Wilmering, T. Semantic description of timbral transformations in music production. In *ACM Multimedia* (2016).

[178] Stables, R., Enderby, S., Man, B., Fazekas, G., and Reiss, J. D. Safe: A system for the extraction and retrieval of semantic audio descriptors. In *Proceedings of the International Society for Music Information Retrieval Conference* (2014).

[179] Stasis, S., Stables, R., and Hockman, J. A model for adaptive reduced-dimensionality equalisation. In *Proceedings of the International Conference on Digital Audio Effects* (2015), vol. 30.

[180] Stasis, S., Stables, R., and Hockman, J. Semantically controlled adaptive equalisation in reduced dimensionality parameter space. *Applied Sciences 6*, 4 (2016), 116.

[181] Stowell, D. *Making music through real-time voice timbre analysis: machine learning and timbral control*. PhD thesis, Queen Mary University of London, 2010.

[182] Sundaram, S., and Narayanan, S. Vector-based representation and clustering of audio using onomatopoeia words. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2006).

[183] Sundaram, S., and Narayanan, S. Audio retrieval by latent perceptual indexing. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (2008).

[184] Tavana, A. Democracy of sound: Is garageband good for music?, 2015.

[185] Theberge, P. *Any sound you can imagine : making music/consuming technology*. Music/culture. Wesleyan University Press : University Press of New England, Hanover, NH, 1997.

[186] Thurstone, L. L. The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology 21*, 4 (1927), 384–400.

[187] Titze, I. R. *Principles of voice production*. National Center for Voice and Speech, Iowa City, Ia, 2000.

[188] Toulson, E. A need for universal definitions of audio terminologies and improved knowledge transfer to the audio consumer. In *Proceedings of the Art of Record Production Conference* (2006).

[189] Tsukida, K., and Gupta, M. R. How to analyze paired comparison data. Report, University of Washington, 2011.

[190] Varse, E. *The Liberation of Music*. Audio culture: Readings in modern music. Continuum, New York, 2004.

[191] Vertegaal, R., and Bonis, E. Isee: An intuitive sound editing environment. *Computer Music Journal 18*, 2 (1994), 21–29.

[192] Vincent, E., Araki, S., and Bofill, P. *The 2008 signal separation evaluation campaign: A community-based approach to large-scale evaluation*. Springer, 2009, pp. 734–741.

[193] Vincent, E., Gribonval, R., and Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing 14*, 4 (2006), 1462–1469.

[194] Vincent, E., Sawada, H., Bofill, P., Makino, S., and Rosca, J. *First Stereo Audio Source Separation Evaluation Campaign: Data, Algorithms and Results*, vol. 4666 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, book section 69, pp. 552–559.

[195] Wang, D., and Brown, G. J. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[196] Wessel, D. L. Timbre space as a musical control structure. *Computer Music Journal 3*, 2 (1979), 45–52.

[197] Williams, E. J. The comparison of regression variables. *Journal of the Royal Statistical Society. Series B (Methodological) 21*, 2 (1959), 396–399.

[198] Wold, E., Blum, T., Keislar, D., and Wheaten, J. Content-based classification, search, and retrieval of audio. *IEEE MultiMedia 3*, 3 (1996), 27–36.

[199] Yee-King, M., and Roth, M. Synthbot: An unsupervised software synthesizer programmer. In *Proceedings of the International Computer Music Conference* (2008).

[200] Yee-King, M. J. *Automatic sound synthesizer programming: techniques and applications*. PhD thesis, Univ. of Sussex, 2011.

[201] Yumer, M. E., Chaudhuri, S., Hodgins, J. K., and Kara, L. B. Semantic shape editing using deformation handles. *ACM Transactions on Graphics 34*, 4 (2015), 86.

[202] Zacharakis, A., Pastiadis, K., Papadelis, G., and Reiss, J. An investigation of musical timbre: Uncovering salient semantic descriptions and perceptual dimensions. In *Proceedings of the International Society for Music Information Retrieval Conference* (2011).

[203] Zannoni, M. Approaches to translation problems of sensory descriptors. *Journal of sensory studies 12*, 3 (1997), 239–253.

[204] Zhang, Y., and Duan, Z. Retrieving sounds by vocal imitation recognition. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing* (2015), IEEE, pp. 1–6.

[205] Zhang, Y., and Duan, Z. IMISOUND: An unsupervised system for sound query by vocal imitation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2016).