Interactive Learning for Creativity Support in Music Production

Mark Cartwright
Northwestern University
2133 Sheridan Rd, 3-204
Evanston, IL 60208 USA
mcartwright@gmail.com

Bryan Pardo
Northwestern University
2133 Sheridan Rd, 3-323
Evanston, IL 60208 USA
pardo@cs.northwestern.edu

Abstract
Synthesizer programming interfaces are usually complex, discouraging novice users from exploring timbres outside the confines of "factory presets". This paper presents an alternative approach to synthesizer interfaces to enable novices to quickly find their target sound in the large, generative timbre spaces of synthesizers, while also allowing for exploration.

Keywords
Timbre, music production

ACM Classification Keywords
H.5.1 [Information Interfaces and Presentation (e.g. HCI)] Multimedia Information Systems---audio input/output; H.5.2 [Information Interfaces and Presentation (e.g. HCI)] User Interfaces---interaction styles; H.5.5 [Information Interfaces and Presentation (e.g. HCI)] Sound and Music Computing---*Signal analysis*, *synthesis*, *and processing*, *Systems*

Introduction
Timbre is attributes of a sound that allow us to distinguish two sounds of the same pitch and volume as distinct [2]. An important aspect of making music in the 21st century, music production is the process of manipulating and/or synthesizing the timbre of a sound. Unfortunately, as music production tools have become more advanced, their interfaces have become more complicated. This is particularly true of software-based sound synthesizers For example, Apple Inc.'s ES2 synthesizer has 125 controls. Even if those controls were simply binary switches, the control space would consist of 2^{125} ($\approx 10^{38}$) combinations. Such tools require years of experience with sound synthesis to understand the controls and use them effectively (i.e. they are high threshold interfaces). They often have many parameters whose meanings are unknown to most novices (e.g. "LFO1Asym" on the ES2). For many musicians, this translates into the inability to actualize one's ideas due to the barrier of the interface. Even for experienced users, the tedium of these interfaces takes them out of their creative flow state, hampering productivity. Although simpler interfaces do exist (e.g.

Apple Inc.’s GarageBand), they lack the flexibility of the complex interfaces, resulting in a small timbre palette constructed of presets, templates, and parameters with few creative options (i.e. they are low-ceiling interfaces). These interfaces stifle creativity, squelching ideas that are outside of the preset confines.

We need interfaces for music production that allow even someone with little experience to utilize existing powerful music production algorithms while maintaining the flexibility of the complex interfaces. We need to design music production interfaces that uphold the important design principles for creative thinking: low threshold, high ceiling, wide walls [6] While many of the ideas presented in this paper can be applied to interfaces for other types of music production, we will focus on synthesizer programming here.

Related Work

Researchers have approached this problem several different ways. For example, some researchers have sought perceptual dimensions of timbre to be used as control parameters of synthesis algorithms [7,8]. Their idea was that these higher-level control parameters would be directly linked to perceptual dimensions, providing considerable control over the perceived dimensions of sound using only a few dimensions in the control space.

However, while providing more relevant dimensions for control is an improvement, as noted earlier, the timbre spaces of synthesizers are overwhelmingly large. To overcome these large spaces, there have been many approaches in which the machine breaks down the search space into smaller, more digestible spaces. For example, researchers have attempted to utilize evolutionary computing (EC) algorithms to search a timbre space for a particular sound [1,4,9]. In [1] and [4], the users performed the fitness evaluation themselves and start with a random population rather than providing an initial target signal. Since EC algorithms take tens or hundreds of generations to converge on the objective [9], those two approaches suffer from a fitness bottleneck due to the time required for a human to evaluate dozens of sounds at each generation. For this reason, such approaches are limited to the exploration of the synthesis timbre space rather than searching for a particular timbre. To circumvent this problem, some researchers [5,9] have implemented optimization-based approaches in which the user provides a target audio example and the objective function is instead evaluated by the machine. These approaches are formulated as minimization problems in which the variables being solved for are the synthesis parameters and the objective function is a distance function between a target audio example and the audio output of the synthesizer.

However, these optimization-based approaches make some assumptions that may not hold in realistic situations. For example, they assume that the user has a recording of the exact sound they are searching for, but it's more likely that they have one or more recordings of sounds that share some of the characteristics of the sound they are looking for but not all of them. They also assume that one measure of timbre similarity will hold for all users. However, timbre similarity is user dependent [3] with users weighting some features more than others, etc. Additionally, these approaches cannot account for when the user’s objective evolves over time. This may occur once the user has heard more sounds in the search process.

Their concept of what they are searching for becomes clearer and more defined as time progresses [1].

An Interactive Learning Approach

The first author is an experienced computer musician who has frequently collaborated with less technically experienced musicians to synthesize sounds and has observed the following interaction in multiple occasions. To describe the desired sound, one person begins by providing one or more example recordings or by mimicking the sound through vocalization. When multiple examples are given, each may indicate different constraints on the desired sound. This scenario often results in an iterative process of successive approximations in which the experienced user synthesizes a sound; the other person provide feedback; another sound is synthesized; more feedback is given; and so on until they are satisfied.

What if users were able to interact with a software-based synthesizer in a similar way? By providing examples, listening and giving feedback, the users would focus on the sound rather than fretting over the user interface of the synthesizer. In addition, by reducing the time musicians spend on tedious production tasks, they may increase their productivity by maintaining their flow and increasing the time spent creating original music. Such a tool would support creativity rather than stifle it. In this workflow, the software would essentially act as a production assistant, performing the majority of the mundane, tedious tasks, while the user still maintains creative control over the music production by guiding the assistant in the process.

By bringing the user back into the loop, the deficiencies of the optimization-based approaches would be addressed. With the feedback from each iteration, the objective function could be updated to match the user's timbre preferences, learning their user specific timbre similarity function over time. In addition, by having successive iterations with feedback, the initial examples provided by the user could be “soft” targets since more information would be obtained from the user than that of simply one target example. The import dimensions of the initial example audio would be learned through the feedback.

Such an approach could also overcome the deficiencies of the human evaluation-based EC based approaches that take a long time to converge for specific targets. Since the user would be able to provide an initial audio example (for which they do not need to know the synthesis parameters), the number of iterations until convergence could be drastically reduced, and the interface could support both the target based use case as well as an exploratory use case.

Challenges

There are of course many challenges to address when designing and implementing such an interface. The first is to determine what kind of feedback the user should return to the software and a machine learning mechanism to learn from these few samples of feedback. Fortunately, these questions have motivated many papers in the field of interactive content-based image retrieval (CBIR) [10]. The CBIR community's research on interactive search algorithms utilizing relevance feedback and active learning may prove a good starting point. However, as mentioned earlier, the size of the timbre space of an audio synthesizer is

enormous. It is much too large to simply sample the entire space and treat it as a library as in the CBIR approaches. We therefore need to determine 1) how to compensate when using a sub-sampling of the space, 2) how coarsely we can sub-sample the space, and 3) how to most efficiently choose those samples. Lastly, when accommodating both target based search and exploratory search, there is a trade-off when presenting results for the user to give feedback on. The software needs to balance the results that are most likely the target, the results whose feedback will be most informative, and the results that will most effectively allow the user to explore the synthesis timbre space.

Conclusions

Current interfaces for programming synthesizers are either too limiting or too complex for an efficient workflow. In this paper we proposed that by designing audio interfaces in which users provide audio examples, listen to suggestions and give feedback, users can focus on the sound rather than low-level synthesis parameters. We are currently exploring this direction and developing such a system.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0824162, and in part by National Science Foundation Grant 0757544.

References

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

[2] American National Standards Institute. *American national psychoacoustical terminology*. American Standards Association, New York, 1973.

[3] McAdams, S. Perspectives on the Contribution of Timbre to Musical Structure. *Computer Music Journal*, 23, 3 1999), 85-102.

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

[5] Mintz, D. *Toward timbral synthesis: a new method for synthesizing sound based on timbre description schemes*. University of California, Santa Barbara, 2007.

[6] 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.

[7] Vertegaal, R. and Bonis, E. ISEE: An Intuitive Sound Editing Environment. *Computer Music Journal*, 18, 2 (1994), 21-29.

[8] Wessel, D. L. Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3, 2 (1979), 45-52.

[9] Yee-King, M. and Roth, M. Synthbot: An unsupervised software synthesizer programmer. In *Proc. of ICMC* (2008).

[10] Zhou, X. S. and Huang, T. S. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8, 6 (2003), 536-544.