

END-USER DEVELOPMENT OF SONIFICATIONS USING SOUNDSCAPES

KatieAnna Wolf, Genna Gliner

Department of Computer Science
Princeton University
Princeton, NJ 08540

kewolf genna @university.edu

Rebecca Fiebrink

Department of Computing
Goldsmiths, University of London
London, UK

r.fiebrink@gold.ac.uk

ABSTRACT

Designing sonifications requires knowledge in many domains including sound design, sonification design, and programming. Thus end users typically do not create sonifications on their own, but instead work with sonification experts to iteratively co-design their systems. However, once a sonification system is deployed there is little a user can do to make adjustments. In this work, we present an approach for sonification system design that puts end users in the control of the design process by allowing them to interactively generate, explore, and refine sonification designs. Our approach allows a user to start creating sonifications simply by providing an example soundscape (i.e., an example of what they might want their sonification to sound like), and an example dataset illustrating properties of the data they would like to sonify. The user is then provided with the ability to employ automated or semi-automated design of mappings from features of the data to soundscape controls. To make this possible, we describe formal models for soundscape, data, and sonification, and an optimization-based method for creating sonifications that is informed by design principles outlined in past auditory display research.

1. INTRODUCTION

“An important goal of sonification is to yield an auditory display that will be intuitively maximal in meaning to the observer [1].” This statement from Gregory Kramer was published during the early years of the auditory display community and stresses the importance of considering the end user (listener) when designing sonifications. Each end user of a sonification system may have different objectives, data, perceptual abilities, and aesthetic preferences, so we should allow users to develop auditory interfaces that best fit their needs and desires.

Designing these displays is a complex task that involves knowledge of the particular data to be represented, the design of the sounds, the best way to convey the data using sound, and the programming ability to build the system. Typically, this knowledge is distributed between end users, who are usually experts in the data and know their intentions for using the system, and system designers, who know about sound and sonification design. Iterative refinements of designs between end users and designers have been proposed as one way to balance the technical knowledge needed to develop a system with practical knowledge of how

the system will be used [2]. However, in many cases once a system is created and deployed, the end user no longer has the opportunity to make changes to the system.

We are interested in putting the end user in control of the design process and assisting them as they iteratively design their own sonifications and explore the space of possible sonifications. Our work applies methods and techniques from the field of human-computer interaction (HCI) to the design of sonifications. In particular, this paper presents a novel approach for sonification design that encompasses:

1. Models as structures for defining components of the system: a soundscape model for defining the sound structures, a data model for defining the properties (i.e. features) and structures of the data, and a sonification model that utilizes the soundscape and data models as well as user input to create a sonification;
2. A policy as a part of the sonification model for defining the interaction between the user, the data, and the sound, and that encapsulate past knowledge of sonification design on how to best convey data with sound; and
3. An interaction workflow that describes how users interact with the system, and how that interaction is used to guide the creation of the sonifications.

Our approach is embodied in a prototype system we developed called the Environmental Soundscape Creator (ESCaper). ESCaper currently demonstrates sonification of two different data sources and allows end users to create, generate, and refine sonifications using a graphical user interface. The goal of ESCaper is to demonstrate an implementation of our approach and allow us to evaluate our design decisions in practice (which we hope to do in future work).

This paper is structured as follows: we first motivate our approach by relating it to previous work done in HCI and sonification, and we describe how the work in this paper builds off of our own previous work in this area. In Section 3, we then present our models for structuring and defining soundscapes, data, and sonifications. In Section 4 we present our policy for creating data-to-sound mappings from user-specified example soundscapes and data. Section 5 describes the use of our models, policy, and interaction workflow used in ESCaper as it applies to data taken from Twitter and the stock market.

2. MOTIVATIONS

In this section we present related work in HCI focused on supporting users in the design process, as well as work in auditory displays



This work is licensed under Creative Commons Attribution Non-Commercial 4.0 International License. The full terms of the License are available at <http://creativecommons.org/licenses/by-nc/4.0>

that utilizes interactivity and soundscapes in sonification. We also describe the relationship of our recent work to this paper.

2.1. Related Work

2.1.1. Human-Computer Interaction

Several design methodologies, such as user-centered design and participatory design, incorporate the user in the design process. In both of these methodologies, users give feedback in the design phases of a project as designers iteratively refine designs with end users. However, once these designs are finalized and deployed, making changes over time is not supported. Therefore the design approach called *metadesign* was created to open up solution spaces for users to explore rather than giving them complete solutions [3]. Our sonification approach utilizes metadesign by presenting end users with a space to explore possible sonifications, instead of giving them a completed sonification system.

However, the sonification design space is very large and may be tedious or difficult to explore for a novice sonification designer. We can therefore look at specific approaches in HCI to assist novice designers in exploring the design space. For instance, some interfaces incorporate expert knowledge to automatically generate designs (e.g. automatically generating magazine covers [4], web designs [5], and user interfaces [6]). Another way to interactively create and explore designs is to use examples supplied by the end user. Examples are described as “a cornerstone of creative practice” [7] and give users a simple way to communicate their design goals. The *Bricolage* algorithm in particular focuses on transferring the design (e.g., layout) of a web page selected by the end user onto the content of another web page [5]. Our approach to sonification design incorporates both these strategies, combining user-specified examples with known sonification principles to automate aspects of the design process.

Other relevant work in HCI has proposed methods for end-user creation of cross-modal mappings. Fiebrink et al. enabled musicians to create mappings from gesture to sound using supervised learning, where users iteratively provided examples of gestures and matching sounds [8]. That work demonstrated instantiating gestural control systems from user-provided examples, helped users abstract away low-level details, sped up the mapping exploration process, and facilitated use by non-programmers. Here we employ a similar user interaction paradigm to create cross-modal mappings from data to sound using examples. However, we use a novel mapping generation approach that incorporates sonification design principles, rather than using general-purpose supervised learning algorithms.

2.1.2. Interactivity in Sonification

The triennial Interactive Sonification Workshop (ISon) started in 2004 focuses on putting sonification users at the heart of an interactive control loop. Some of the ways that this can be done include: using real-time sonifications of actions produced by users, applying user-centered design to the development of sonification systems, and allowing users to interact with a playback of a data display. However, little work has focused on putting end users at the center of designing sonifications. Some toolkits have been developed that allow users to input data, select mappings, and control the playback of sonifications [9, 10], but these use complex interfaces with terminology that may not be intuitive for a novice sound designer. Additionally, the creation of good mappings may not be

easy for someone who does not have experience in sonification design. Our goal is to make sonification design available to end users who have little or no sound or sonification design experience. We do this by providing methods for automatic generation of designs using known principles for sonification and auditory display.

Hermann et al. had similar motivations in their design of an interactive technique for automatically generating sonifications. They used a listener’s relevance feedback about the perceptual quality of proposed sonifications to guide in the creation of an optimal sonification [11]. We also utilize user input in the creation of our sonifications, but we do so in multiple ways. Users of our system can directly select the sounds used in the sonification, specify details of how a sonification should be created, and continually develop and reapply their designs to the data.

2.1.3. Soundscapes in Sonification

Some of the ways that soundscapes have been used in the field of auditory display include using ambisonic sound recordings from urban soundscapes as a layer in a display [12], synthesizing “tweetscapes” using data from Twitter [13], and creating soundscapes of complementary ecological sounds by using sampled audio segments as units to build a display [14]. All of these uses of soundscapes aim to create a sound environment where the individual streams of sounds fit together cohesively to form an immersive environment, while also conveying information about the data.

Our models and mapping policy presented here are designed to support natural acoustic soundscapes that include animal vocalizations and sounds of the weather. These soundscapes have been identified as useful for sonification as they can be easily distinguished from the background and have even been found to be “relaxing” [14] with the potential to be less fatiguing than other sound interfaces [15]. While our models and mapping policies could be generalized to sonifications that use any type of sounds (e.g., music or other types of recordings), we use natural sounds from the same physical location that have evolved in a way specific to that location (i.e. each sound has a unique timbre that can be easily distinguished from others in that same environment). This difference in timbre can be identified by everyday people using terms that they know such as *wolf howling* and *owl hooting*, rather than using sound terminology like *frequency* and *timbre*.

2.2. Building on Recent Work

Our recent work [16] has presented formal models for soundscapes, data, and sonifications, and demonstrated a proof-of-concept sonification of Twitter data. We directly build upon that work here by describing the relationship of our model formulations to past work in the auditory display community. We first present a modified and expanded mapping policy that explicitly incorporates user interaction. Users are able to specify their data and sound selections, and our updated policy allows users to guide the generation of mappings to produce sonifications that either prioritize conveying properties of the data or that prioritize producing output sounds that are most similar to the initial example soundscape. Finally, in this work we present a prototype application that demonstrates the interaction workflow of our approach and the use of our models with the new policy for two different datasets.

3. MODELS

We define formal models of soundscape, data, and sonification that are amenable to algorithmic control in order to make it possible to automate the sonification design process in a user-friendly way. A soundscape model describes the structural hierarchy of a soundscape to make it easier to analyze, understand, and utilize. For those same reasons, we create a data model to provide a way of structuring the data in a dataset. Our sonification model is a framework that defines the data-sound mapping, which takes an input soundscape and a dataset, and utilizes a policy to analyze the structure of those models and incorporate user interaction to produce an output soundscape that represents a sonification of the data:

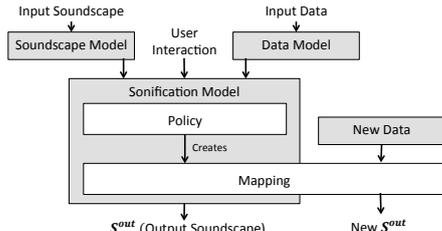


Figure 1: Here we depict our model formulations.

3.1. Soundscape Model

3.1.1. Soundscape Description

We define a soundscape as a sequence of soundscape events $S_t \in \mathcal{S}$. Each soundscape event has a finite set of possible sound groups $G_j \in S_t$, and each sound group consists of a finite number of sound recordings $g_{j,k} \in G_j$ called sound samples. For example, in Figure 2 each distinct bird chirp recording is a separate sound sample ($g_{2,1} \dots g_{2,4}$), which is a part of the bird sound group G_2 . Every soundscape, sound group, and sound sample has a set of properties, called *features*, described below and listed in Table 1.

3.1.2. Soundscape, Sound Group, Sound Sample

Two types of soundscapes are relevant to our approach: an *input soundscape* (denoted \mathcal{S}^{in}) is the pre-recorded soundscape provided by a user as an example of the sonification he or she might like to create from data. An *output soundscape* (denoted \mathcal{S}^{out}) is a soundscape that is generated by a specific sonification design (i.e., a mapping) in response to a specific dataset. Both types of soundscapes can be characterized by features listed at the left of Table 1). Furthermore, we can characterize each sound sample and sound group in terms of sample- or group-level features, respectively.

Our approach of using soundscapes for sonification enables us to encode data using sample recordings of real-world sounds, or what Kramer describes as “realistic voices” [1]. Kramer describes six dimensions of recordings that can be manipulated to convey properties of data: speed (or density), duration, envelope, master

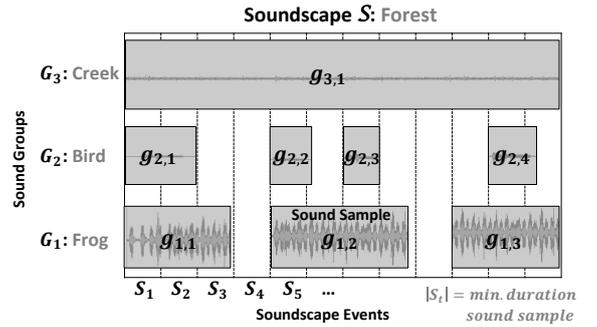


Figure 2: An example forest soundscape model with three sound groups G_j , each with a different number of sound samples $g_{j,k}$.

loudness, master pitch (pitch area), and master brightness. Inspired by that work, we define eight features of a sound sample (found in center of Table 1) that can be analyzed (in an input soundscape) or manipulated in response to data (in an output soundscape). One of the key actions of our sonification policy discussed in Section 4 is to choose which features of the sound samples in the output soundscape will convey the data. We hope to explore additional sound features (e.g. reverb and more complex spatial representations) to take advantage of other human perceptual abilities.

A sound group has features that aggregate information about the sound samples in that group, which are depicted at the right of Table 1. For instance, the ‘on/off’ feature denotes whether any sound sample occurs in the sound group. The ‘on/off’ feature can thus change over each soundscape event. The *type* of a sound group is either *interval* or *instant*, depending on the duration and sonic properties of the sound group. Interval sounds are those that occur continuously throughout the soundscape, while instant sound groups have multiple samples occurring at varying intervals. We use a basic heuristic for defining whether a group in an input soundscape is interval or instant. If a sound group has only a single sound sample that has a duration equal to the duration of the entire soundscape (e.g. a river flowing), then that sound group is interval, else it is instant (e.g. a bird chirping).

3.2. Data Model

3.2.1. Data Description

Our automated mapping policy requires a model for the data that is to be sonified. We consider a dataset \mathcal{D} to be made up of a finite set of *data points* $d_{i,t}$. Data points that share a common feature (i.e. the source of the information) are grouped into a *data group* D_i . Similarly, data points that occur at the same time (or within a small time interval) are also collected in a finite set called a *data event* E_t . The duration of a data event is equal to the minimum interval between any two adjacent data points. Figure 3 shows a visual representation of our data model.

Soundscape Features	Value Range
Soundscape Type	Input, Output
Num. of Sound Groups	[0, max]
Total Num. of Sound Events	[0, max]
Duration of Soundscape Event	Min. sample duration
Duration of Soundscape	[0, max]

Sound Sample Features	Value Range
Duration of Sample	[0, max]
Start of Sample	[0, max]
Spectral Features	[min, max]
Sample On/Off	0, 1
Gain	[0, 1]
Panning	[-1, 1]
Layering	[0, 6]
Playback Rate	[0.5, 1.5]

Sound Group Features	Value Range
Type	Interval, Instant
% of Soundscape Duration	[0, 100]
Num. of Samples in Group	[0, max]
Avg. Spectral Features	[min, max]
Group On/Off	0, 1

Table 1: The features used to describe a soundscape, sound group and sound sample, along with their value ranges.

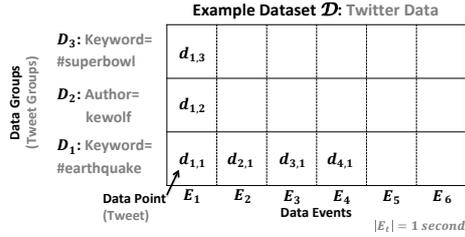


Figure 3: An example of our data model with a Twitter dataset \mathcal{D} containing data points (tweets) $d_{i,t}$ that each belong to a data group (tweet group) D_i and a data event E_t . In data event E_1 a tweet occurred in all three groups, but in $E_2 \dots E_4$ only tweets containing #earthquake occurred.

3.2.2. Data Features

Each data point and data group contain a set of features which are dependent on the domain of the data. However, we have defined some requirements of all datasets and guidelines for developing the data group and data point features.

Alberto de Campo wrote that “To ensure that the auditory gestalts of interest will be easily perceptible, the most fundamental design decision is the time scale [17]”. In our approach, each dataset must have a way to represent an ordered sequence of their data points, i.e. a feature to represent time. Since humans are particularly sensitive to perceiving changes in sound over time [18], we want to create sonifications that “display” an aspect of the data using these perceived temporal changes. Each data point may or may not contain information, for instance one data group may have information every four seconds while another captures information each second, and still another captures data sporadically. To determine the window for a single data event, we use the minimum time difference between data points. This value should be represented in terms of seconds, however, as de Campo points out in his Data Sonification Design Space Map, gestalts of 100 ms and less are difficult to discern meaningful information, while longer than 30 seconds take great concentration [17]. Thus we stick to de Campo’s suggestion of using 1–3 seconds for a time frame for single gestalts (i.e. for the duration of our data events) by adjusting our time values over our dataset to fit into this range.

As each dataset has its own particular features, we only propose a generalized model for defining these features. However, in Section 5 we apply this model and feature representation to real-world datasets. One required feature of all data groups is the *type* feature. There are two types of data groups: *single* data groups, where each data point $d_{i,t} \in \mathcal{D}_i$ is represented as its own discrete event, and *aggregated* data groups, that aggregate an aspect of the data point features. The equation we use to determine the data group type uses the number of data points per second (DPPS) and the shortest duration sample in the soundscape (SD):

Aggregated Data Group: $DPPS \times SD \geq 0.5$

Single Data Group: $DPPS \times SD < 0.5$

We define three categories of data point features:

Time: Portrayed in terms of seconds.

Source: Describes where the data point came from (this can be consistent across many data points).

Single Data Value (SDV): Information from the source at a specific time.

Along with *type*, we define two other group feature categories:

Type: Representation of the data points (aggregated or single)

Filter: Describes the commonality between the data points in the group (e.g. from the source, some SDV, etc.).

Aggregated Data Value (ADV): Aggregates information from data points in the group (e.g. a minimum, maximum, or average value of a SDV) and normalizes it.

3.3. Sonification Model

Our sonification model is motivated by a model created by Hermann et al. [11] and is built upon our original model defined in [16]. We formally define the sonification model as the mapping $[\mathcal{D}, \mathcal{S}^{in}] \mapsto \mathcal{S}^{out}$ that takes the dataset \mathcal{D} and an input soundscape \mathcal{S}^{in} and outputs the resulting soundscape \mathcal{S}^{out} . We define the output soundscape as the sequence of all sonification events $\mathcal{S}^{out} = [S_1, S_2, S_3, \dots, S_n]$ where each of these sonification events is defined by:

$$S_t(\mathcal{D}) = \sum_{d_{i,t} \in E_t} \phi(d_{i,t}, \theta(\mathcal{S}^{in}, \mathcal{D})). \quad (1)$$

Here $\phi(d_{i,t}, \theta(\mathcal{S}^{in}, \mathcal{D}))$ represents an acoustic event occurring at time t . This acoustic event is created based upon two functions (denoted ϕ and θ and defined in the next section), which determine how a data point at time t will be represented using sounds from the input soundscape \mathcal{S}^{in} .

4. MAPPING GENERATION POLICY

We present one possible policy which is simply a function that determines the mappings between the data and the sound. We derive our policy from previous work in the field of auditory display. For instance, Kramer [1] states that “...important features will be more discernible if the display is structured to reflect structures in the data.” Using this idea, we take advantage of the parallel structures in our data and soundscape to create sonifications where:

1. Each selected data group is mapped to a sound group ($D_i \mapsto G_j$). The group mapping function θ determines which data group should be mapped to which sound group.
2. Each data point in a data group ($d_{i,t} \in D_i$) is mapped to an individual sound sample in the corresponding sound group ($g_{j,k} \in G_j$). The point-sample mapping function ϕ determines how the data points should be represented by the sound samples.

When a user interacts with the system, they select a tuning parameter to choose whether they wish to have the system create mappings such that the output soundscape is either: as similar to the input as possible or designed to best convey the data. They can then select the data they wish to sonify and either manually select the sound group to which the data will be mapped, or they can allow the system to automatically choose one for them (determined by the group mapping function θ).

4.1. Group Mappings

By mapping each data group to a sound group, we are matching the parallel structures in our models. Similarly, for data that will be updated as it is aggregated continuously over time, we want to represent it with a sound that can be played over long intervals

(an aggregated-interval mapping). When the data is displayed for every single data point we want the sound to be played at a specific instant (a single-instant mapping). This is not only supported by Kramer’s similarity of structure rule [1], but also by Ahmad et al. who proposed that instant-based temporal information (in our case “instant” sounds) are used to specify a point in time, where as interval-based temporal information (“interval” sounds) are used to indicate status or progress [19].

A dataset is defined as $\mathcal{D} = \mathcal{D}^{sel} \cup \mathcal{D}^{avail}$ where \mathcal{D}^{sel} is the set of user selected data groups to use in our sonification, and \mathcal{D}^{avail} is the set of data groups that have not been selected. The selected data is also a union of two sets: $\mathcal{D}^{sel} = \mathcal{D}^{sel,m} \cup \mathcal{D}^{sel,\bar{m}}$, user selected data groups that are mapped (i.e. there is a data-sound mapping that uses this data group), and those that selected by the user, but are not mapped. A soundscape is similarly defined as the union of sound groups that have been mapped (i.e. there is a data-sound mapping that uses this sound group) and those that have not: $\mathcal{S} = \mathcal{S}^m \cup \mathcal{S}^{\bar{m}}$.

The group mapping function selects specific sound groups to which each user-selected data group should be mapped: $\theta : \mathcal{D}^{sel} \mapsto \mathcal{S}^m$. These mappings are constrained to follow the aggregated-interval and single-instant rules described above such that: $\mathcal{D}^{sel,\bar{m},aggr} \mapsto \mathcal{S}^{\bar{m},inter}$ and $\mathcal{D}^{sel,\bar{m},single} \mapsto \mathcal{S}^{\bar{m},inst}$. Given all possible permutations we can then evaluate each mapping $\theta_i \in \Theta$ and use a quality function Υ to evaluate and select the best one. Our quality function is a weighted sum of two quality terms. The soundscape similarity quality term Υ_{ss} evaluates a mapping based on how similar the output soundscape is to the input soundscape, and the data conveyance quality term Υ_{dc} evaluates a mapping based on how well the data is conveyed in the output soundscape. The weighting of these quality terms is input by a user as $\alpha \in [0, 1]$.

$$\theta = \arg \max_{\theta_i} \alpha \Upsilon_{ss}(\theta_i) + (1 - \alpha) \Upsilon_{dc}(\theta_i) \quad (2)$$

4.1.1. Data Conveyance Quality Term

As reported by de Campo, “After time scale, the number of streams is the second most fundamental perceptual design decision [17].” Thus in evaluating a mapping where our main goal is to represent the data as clearly as possible, we want to make sure we are not over-burdening the user with too much information. A recent study found that listeners could accurately identify five concurrent variables presented together in a single sonification [20]. We utilize that principle and limit the number of data groups that a user can select to five ($|\mathcal{D}^{sel}| \leq 5$). We also restrict the number of aggregated-interval mappings to three ($|\mathcal{D}^{sel,aggr}| \leq 3$), as we currently have three possible parameter choices for aggregated-interval mappings. By reducing the data a user is allowed to select, we are also reducing the number of possible permutations we evaluate in our data conveyance quality term.

We use additional principles outlined by Kramer on how to use sounds to make the data more distinguishable in the sonification [1]. He suggests using distinctly different timbres to distinguish different data groups (or “families”) to make the information in auditory displays more perceptible. We incorporate this by using recorded sounds from a soundscape and choosing sound groups in our mapping that have a diverse range of spectral centroids (a measure of the “brightness” of a sound). Kramer also mentions that the use of sounds with a minimum overlap in frequency can be helpful to avoid masking, so we choose sound groups by maximizing the distance between their frequency representations, for

example the Mel-frequency cepstrum coefficients (MFCC) or FFT bin values. In our implementation we take the L2 norm between the power spectrum bins. We use a weighting term β to adjust the importance of including diverse timbres versus minimizing overlapping frequencies. Let b_i be the average spectral centroid (over all sound samples in a group) for each sound group in \mathcal{S}^m and \vec{f}_i be a vector representation of the average frequency components (over all sound samples in the group) for each sound group in \mathcal{S}^m . Then the data quality term is calculated as follows:

$$\Upsilon_{dc}(\theta_i) = \sum_{i=1}^{|\mathcal{S}^m|} \sum_{j=1}^{|\mathcal{S}^m|} \beta \|b_i - b_j\|^2 + (1 - \beta) \|\vec{f}_i - \vec{f}_j\|^2. \quad (3)$$

4.1.2. Soundscape Similarity Quality Term

To create soundscapes that have a similar sound as the input soundscape, we include sounds in the output soundscape if they occur during more than 75% of the duration of the input soundscape ($p > 75\%$), even if they are not mapped. Thus $\mathcal{S}^{out} = \mathcal{S}^m \cup \mathcal{S}^{\bar{m},p}$, where $\mathcal{S}^{\bar{m},p}$ represents the non-mapped sounds that meet our criteria. We compute the similarity measure depending on the difference in the soundscape and sound group features.

The soundscape feature we use is the maximum number of sound groups played at once, which approximates the density of a soundscape. The sound group features we use are the number of samples played, the percentage of soundscape duration, the maximum number of other sound groups played at the same time, and the other sound groups that occur at the same time. We define \vec{h}_i to be the sound group feature vector for each sound group $S_i \in \mathcal{S}^{in}$, and \vec{h}_o to be the feature vector for the same sound group in the output soundscape $S_i \in \mathcal{S}^{out}$. Also let l_i be the maximum number of sound groups played at once for soundscape \mathcal{S}^{in} and l_o be the same for \mathcal{S}^{out} . Our soundscape similarity quality term is thus:

$$\Upsilon_{ss}(\theta_i) = -\gamma \|l_i - l_o\|^2 - (1 - \gamma) \sum_{i=1}^{|\mathcal{S}^{out}|} \|\vec{h}_i - \vec{h}_o\|^2 \quad (4)$$

4.2. Point-Sample Mappings

The function ϕ (equation 1) is used to determine how each element in a data group is represented by a sound sample in the corresponding sound group determined by θ (equation 2). The heuristic for choosing the point-sample mapping depends on the type of group mapping. Additionally, since the data features are dependent on the dataset, we generalize this model for any type of data using the data group and data point feature categories from section 3.2, and then apply it to two datasets in Section 5.

4.2.1. Aggregated-Interval Heuristic

By definition, an aggregated data group contains many data points. If multiple data points are “on” in a single data event, we want to aggregate information at each time event. For instance, we could consider the number of data points at time t or an average of a data point feature over all data points at time t . Since the data is continuously updated, we employ sound groups that are continuously played in the input soundscape and loop them while updating one feature each time our data changes. For the sound sample features, we apply changes continually over time to either the playback rate, layering (density), or loudness (gain). The other sound sample features are not used since continuously updating panning values may

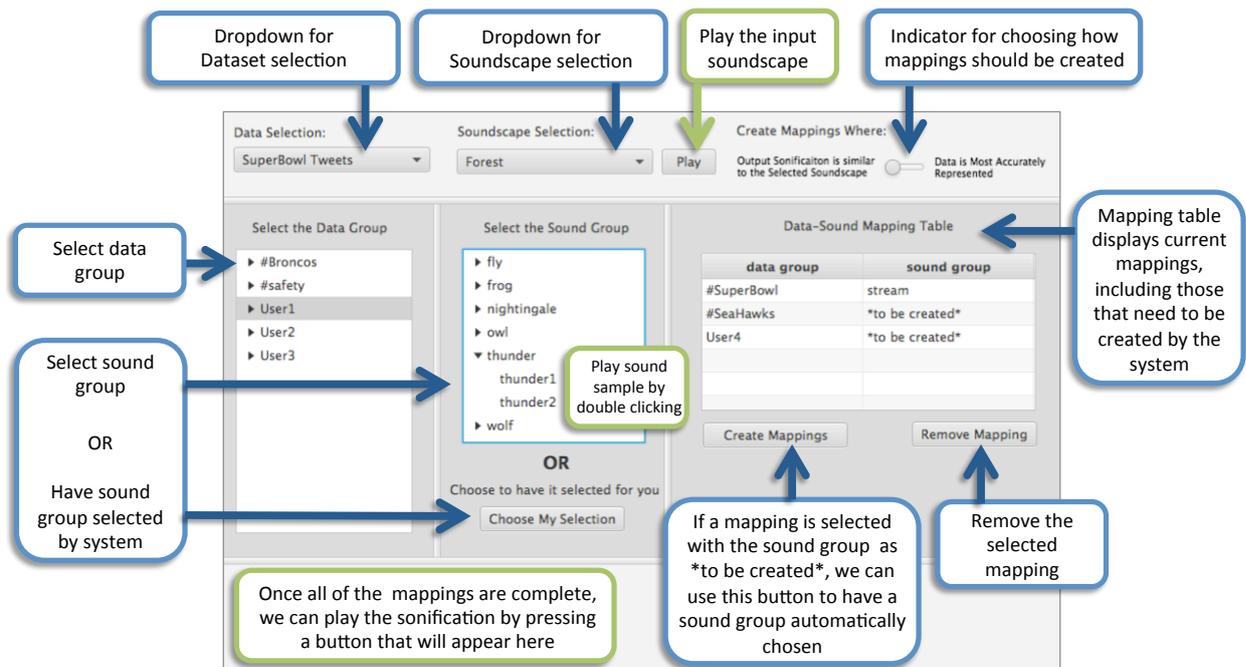


Figure 4: The ESCaper prototype allows users to design sonifications simply by interacting with the interface. There are three different ways the system can produce sound (depicted in the green bubbles): playing the input soundscape, playing the individual sound samples, and playing the sonification.

disorient a user with the constant change of the location of a sound, and changing the duration or start time of a sample does not apply for a continuous sound. Below is a summary of the mappings that we use for updating an Aggregated Data Value (ADV):

Density: $NumberLayers[0, 6] \leftarrow [ADV \times 6]$

Pitch: $PlaybackRate[0.5, 1.5] \leftarrow ADV + 0.5$

Loudness: $Gain[0, 1] \leftarrow ADV$

4.2.2. Single-Instant Heuristic

Instant data represents each data point in the data group, so we use the sound sample features to differentiate each data point. We give each data point a unique sound and location by using the timing and selection of a particular sound and location from the sound group as well as the panning. The Single Data Values (SDV) are information about a single data point and can also be used to influence other dimensions of the sound representation, such as the loudness. To generalize the particular feature to fit any data group, we use the notation $SDV(d_{i,t})$ to represent a SDV feature of any data point. Below is an example formulation of a single-instant heuristic for point-sample mappings:

Attack: $StartTime(g_{j,k})[time] \leftarrow time(d_{i,t})$

Sample Selection: $StartTime(g_{j,k} \in G_j)[time] \leftarrow source(d_{i,t} \in D_i)$

Location: $Panning[left, right] \leftarrow source(d_{i,t})$

Loudness (Optional): $Gain[0, 1] \leftarrow SDV[min, max]$

5. ESCAPER APPLICATION

The Environmental Soundscape Creator (ESCaper) is a prototype system we created to demonstrate our model and policy. Within

ESCaper we apply our model to two datasets: data from the microblog Twitter and data from the stock market. While there is much work to be done in evaluating the policy, interactions, and sonifications of this system, our goal is to demonstrate what a meta-design sonification system might look like, and how we might be able to leverage the knowledge of sonification in assisting end users in the creation of sonifications. In this section we describe the interaction workflow of the system and demonstrate an example of the data features and mapping heuristics.

5.1. ESCAPER Interactions

Figure 4 depicts the graphical user interface for ESCaper. When a user runs ESCaper they are presented with the top pane of the window where they can select a dataset and a soundscape from drop-down menus. Currently, the datasets have been curated for off-line use and are hard-coded into the system. This makes it easier to both compare the same dataset when it is mapped to different soundscapes, and to compare the behavior of different users as they interact with the same data. The input soundscapes are also hard-coded using sound samples from freesound.org.

At this point, the user selects an indicator to describe how they wish to have their mappings created: (1) the output sonification is similar to the selected soundscape or (2) the data is most accurately represented. The main purpose is to determine how the group mappings will be generated (the tunable α in (2)).

A user then selects the data group from the dataset that they wish to sonify. When they click a specific data group the sound group box appears containing the sound groups that are relevant for that data-sound mapping (i.e. interval sound groups appear for aggregated data groups, and instant sound groups appear for single data groups). The user clicks on a particular sound group to create the mapping or they can select to have the system choose the sound group for them. The mapping will then appear in the Data-Sound Mapping Table. If the user selected to have the sound

group chosen for them, the data group will appear with the sound group as “to be created”. The user can repeat this process to create additional mappings, remove the mapping from the table, play the sonification if it does not contain any “to be created” mappings, or have the system generate the “to be created” mappings. To generate the mappings, we use the quality term defined in (2) to compare all possible mappings and select the best one. Once a sonification is running a user can iteratively refine their design by stopping the playback and making changes to the design.

5.2. Twitter ESCaper

Previous work has investigated sonification of Twitter data, including *Tweetscapes* [13] and *I Hear NY4D* [12]. However, neither of these applications allow users to configure the Twitter data selection for their own needs and desires or to directly select the sounds that are used in the Twitter data sonification.

Our Twitter data model adheres to the data model we outlined in section 3.2 and uses features (Table 2) that are slightly modified from our previous work [16]. We apply our aggregated-interval mapping framework to the Twitter data model and use the number of tweets per second (TPS) for the Twitter-specific ADV feature:

Density: $NumberOfLayers[0, 6] \leftarrow [TPS \times 6]$

Pitch: $PlaybackRate[0.5, 1.5] \leftarrow TPS + 0.5$

Loudness: $Gain[0, 1] \leftarrow TPS$

For the single-instant mappings we use two different types of filters to define the tweet groups: the author of the tweet (*author*) and a keyword used in the content of a tweet (*keyword*). If a tweet group has only one author we map the time of each tweet to the start time of a random sample from the sound group:

Attack: $StartTime(g_{j,rand} \in G_j)[t] \leftarrow time(d_{i,t} \in D_i)$

If a tweet group has more than one author, we use the ceiling of the $num.Auth/num.Samp$ to represent the number of source locations to use for panning. If there is only one source location, we do not use panning and simply utilize the sample selection to associate a particular sample with a particular user:

Sample Selection: $StartTime(g_{j,k})[t] \leftarrow author(d_{i,t})$

If there are multiple source locations we use both the playback of a specific sound sample and panning to distribute the sounds spatially around the user and give them a sense of different voices.

Sample Selection: $StartTime(g_{j,k} \in G_j)[t] \leftarrow author(d_{i,t} \in D_i)$

Location: $Panning[left, right] \leftarrow author(d_{i,t})$

5.3. Stockmarket ESCaper

Sonification has been shown to be useful for stock market monitoring [14, 21]. In our Stock Market ESCaper we apply our approach to a stock market dataset that reports price changes for specific stocks and the percentage price change from that stock’s previous close price. These two values were used in a toolkit for stock market sonification along with many others [21]. We only selected two as we seek to demonstrate our approach, rather than present a fully designed sonification system that stock market traders would use.

Each change in the stock price at time t represents a data point, or price change point, $d_{i,t}$. The features for this price change

Tweet Group Features	Feature Type	Value Range
Type	-	Single, Aggregated
Total Num. of Tweets	ADV	[0, inf]
Tweets per Second	ADV	[0, inf]
Num. of Twitter Users	ADV	[1, max]
Author	Filter	“*”, null
Keyword	Filter	“*”, null
Tweet Features	Feature Type	Value Range
Author of Tweet	Source	“*”
Content of Tweet	SDV	“*”
Length of Tweet	SDV	[1, 144]
Time of Tweet	-	[0, max]

Table 2: Twitter data features.

Stock Group Features	Feature Type	Value Range
Type	-	Single, Aggregated
Total Num. of Price Changes	ADV	[0, inf]
Price Changes per Second	ADV	[0, inf]
Num. of Stocks in Group	ADV	[1, max]
Max. Num Stock Change at Once	ADV	[1, max]
Max. % Price Change	ADV	[0, max]
% Change Threshold	Filter	[0, inf], null
Stock Name	Filter	“*”, null
Top 1% Rise in % Price Changes	Filter	0, 1, null
Top 1% Fall in % Price Changes	Filter	0, 1, null
Price Change Point Features	Feature Type	Value Range
Stock Name	Source	“*”
Stock Price	SDV	“*”
% Price Change	SDV	[1, 144]
Time of Price Change	-	[0, max]

Table 3: Stock market data features.

point are the name of the stock whose price changed, the new price of the stock, the percentage change in price from the last closing price, and the time of the price change (bottom of Table 3). Each price change point can be collected into a stock market group D_i which can filter the price points by: a specific stock name (*stock*), a threshold on a particular percentage change in price (*%threshold*), the highest 1% rise in price percentage changes, and the highest 1% fall in price percentage changes.

Similar to the Twitter ESCaper, we use our mapping framework to create the heuristic for the data-sound mappings. For aggregated-interval mappings of the stock market data we use the ADV feature for the normalized percent price change (PPC). The mapping is exactly like that of our twitter aggregated-interval mapping with PPC instead of TPS. For the single-instant mappings, our data source is a specific stock name (*stock*), so we want all price change points of that source to be sonified with the same sound sample. We can use the various stock market group filters to narrow down the price change points that will be sonified.

If a stock market group has only one stock source, then we map the timing of the price change point to a randomly selected sample in the corresponding sound group:

Attack: $StartTime(g_{j,rand} \in G_j)[t] \leftarrow time(d_{i,t} \in D_i)$

If a stock market group has more than one stock source, we use the ceiling of the $num.Sources/num.Samp$ to represent the number of source locations to use for panning. If there is only one source location, we do not use panning, but simply utilize the sample selection to associate a sample with a particular stock:

Sample Selection: $StartTime(g_{j,k})[t] \leftarrow stock(d_{i,t})$

If there are multiple source locations we use both the playback of a specific sound sample and panning to distribute the sounds

spatially around the user and give them a sense of different voices.

Sample Selection: $StartTime(g_{j,k} \in G_j)[t] \leftarrow stock(d_{i,t} \in D_i)$

Location: $Panning[left, right] \leftarrow stock(d_{i,t})$

For both of our ESCaper applications we have presented general formulations to which any Twitter or stock market data set could adhere. We present these formulations to demonstrate how the generalized policy presented in Section 4 can be applied to information particular to a specific data sets.

6. DISCUSSION AND FUTURE WORK

In this paper we built upon our previous sonification model to take advantage of the parallel structures between data and soundscapes. The sonification model presented here uses a more complex policy and optimization technique for the automatic creation of the data-to-sound mappings and is inspired heavily by work in the auditory display community. Additionally, our model allows users to input their preferences into the system (using a simple indicator slider) to guide the automatic creation of mappings. These are our first steps in exploring the possibilities for using interactive systems to assist end users in the design of sonifications.

There are many ways in which we hope to build on this work in the future. We are currently working on user tests to evaluate the algorithms we have presented here, and we will continue to explore more sound features and evaluate the use of our current features in the context of our system. We hope to expand our user interaction to allow users to specify more dimensions of the mapping (polarity, specific point-sample mappings, etc.). Additionally, by using more intelligent schemes such as learning user preferences and analyzing data in real-time to update the mapping, we can make a more practical tool that could help users refine their mappings over time.

7. ACKNOWLEDGMENT

This material is based upon work supported by the NSF GRFP under Grant No. DGE 1148900. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

- [1] G. Kramer, "Some organizing principles for representing data with sound," in *Auditory Display: Sonification, Audification, and Auditory Interfaces*, G. Kramer, Ed. Addison-Wesley, 1994.
- [2] B. N. Walker and G. Kramer, "Mappings and metaphors in auditory displays: An experimental assessment," *ACM Transactions on Applied Perception*, vol. 2, no. 4, pp. 407–412, 2005.
- [3] E. Giaccardi and G. Fischer, "Creativity and evolution: A metadesign perspective," *Digital Creativity*, vol. 19, no. 1, pp. 19–32, 2008.
- [4] A. Jahanian, J. Liu, Q. Lin, D. Tretter, E. O'Brien-Strain, S. C. Lee, N. Lyons, and J. Allebach, "Recommendation system for automatic design of magazine covers," in *Proc. Intelligent User Interfaces*, 2013.
- [5] R. Kumar, J. O. Talton, S. Ahmad, and S. R. Klemmer, "Bricolage: Example-based retargeting for web design," in *Proc. CHI*, 2011.
- [6] K. Gajos and D. S. Weld, "SUPPLE: Automatically generating user interfaces," in *Proc. Intelligent User Interfaces*, 2004.
- [7] S. R. Herring, C. C. Chang, J. Krantzler, and B. P. Bailey, "Getting inspired! Understanding how and why examples are used in creative design practice," in *Proc. CHI*, 2009.
- [8] R. Fiebrink, D. Trueman, C. Britt, M. Nagai, K. Kaczmarek, M. Early, M. Daniel, A. Hege, and P. R. Cook, "Toward understanding human-computer interactions in composing the instrument," in *Proc. Int'l Computer Music Conference*, 2010.
- [9] S. Pauletto and A. Hunt, "A toolkit for interactive sonification," in *Proc. ICAD*, 2004.
- [10] B. N. Walker and J. T. Cothran, "Sonification sandbox: A graphical toolkit for auditory graphs," in *Proc. ICAD*, 2003.
- [11] T. Hermann, K. Bunte, and H. Ritter, "Relevance-based interactive optimization of sonification," in *Proc. ICAD*, 2007.
- [12] B. Boren, M. Musick, J. Grossman, and A. Roginska, "I hear NY4D: Hybrid acoustic and augmented auditory display for urban soundscapes," in *Proc. ICAD*, 2014.
- [13] T. Hermann, A. V. Nehls, F. Eitel, T. Barri, and M. Gammel, "Tweetscapes: Real-time sonification of twitter data streams for radio broadcasting," in *Proc. ICAD*, 2012.
- [14] B. S. Mauney and B. N. Walker, "Creating functional and livable soundscapes for peripheral monitoring of dynamic data," in *Proc. ICAD*, 2004.
- [15] P. Vickers, C. Laing, M. Debashi, and T. Fairfax, "Sonification aesthetics and listening for network situational awareness," in *Proc. Conf. on Sonification of Health and Environmental Data*, 2014.
- [16] K. E. Wolf, G. Gliner, and R. Fiebrink, "A model for data-driven sonification using soundscapes," in *Proc. Intelligent User Interfaces*, 2015.
- [17] A. de Campo, "A data sonification design space map," in *Proc. Ison*, 2007.
- [18] G. Kramer, B. Walker, T. Bonebright, P. Cook, J. H. Flowers, N. Miner, and J. Neuhoff. (1999) Sonification report: Status of the field and research agenda. Prepared for the National Science Foundation by members of the Int'l Community for Auditory Display.
- [19] A. Ahmad, K. Stanney, and H. Fouad, "Theoretical foundations for integrating sound in interactive interfaces: Identifying temporal and spatial information conveyance principles," *Theoretical Issues in Ergonomics Science*, vol. 10, no. 2, pp. 161–186, 2009.
- [20] J. H. Schuett, R. J. Winton, J. M. Batterman, and B. N. Walker, "Auditory weather reports: Demonstrating listener comprehension of five concurrent variables," in *Proc. of Audio Mostly*, 2014.
- [21] F. C. Ciardi, "sMax: A multimodal toolkit for stock market data sonification," in *Proc. ICAD*, 2004.