Technische Universität Berlin

Fakultät I - Geisteswissenschaften
Fachgebiet Audiokommunikation
Audiokommunikation und -technologie M.Sc.

Self-Organizing Maps for Sound Corpus Organization

Master's Thesis

Vorgelegt von: Jonas Margraf
Matrikelnummer: 372625
E-Mail: jonasmargraf@me.com

Erstgutachter: Prof. Dr. Stefan Weinzierl
Zweitgutachter: Dr. Diemo Schwarz
Datum: March 3, 2019

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.
Berlin, den March 3, 2019

.....
Jonas Margraf

Abstract An english abstract.

Zusammenfassung Die Zusammenfassung auch auf Deutsch.

Acknowledgements

This is where the thank yous go.

Contents

1 Introduction

This is the Introduction. Here's a citation about Self-Organizing Maps (SOMs)(Kohonen, 1990).

1.1 Motivation and Problem Description

1.2 Aims and Objectives

1.3 Previous Work

2 Background

The following section intends to provide a theoretical background for two key concepts underlying the work presented in this thesis, namely Audio Feature Extraction and the Self-Organizing Map.

2.1 Audio Feature Extraction

Audio Feature Extraction is the process of deriving *features* from a digital audio signal. A feature represents some sort of descriptive information about the audio data. According to Lerch (2012), this extraction process serves a dual purpose; that of dimensionality reduction as well as a more meaningful representation. A large variety of features for different purposes have been developed (refer to Peeters (2004) for an extensive list as well as Lerch (2012) for an in-depth look at the topic). The following subsections introduce the features used in this work, starting with the pre-processing required to prepare an audio signal for feature extraction and then moving into individual definitions for each feature. The equations presented here are based on the formal definitions given by Lerch (2012) along with the computational implementations of the features in the *Meyda* library for feature extraction in JavaScript (Rawlinson et al., 2015), which in turn adapted the *yaafe* library for Python (Mathieu et al., 2010).

2.1.1 Audio Pre-Processing

Consider a digital audio signal of the form $x[n]$, where n denotes the sample index and $x[n]$ the value of the individual sample at that index.

2.1.1.1 Normalization In order to have a standardized maximum amplitude of 1 across all audio signals, they are normalized such that

$$x_{norm}[n] = \frac{x[n]}{\max x[n]}. \tag{1}$$

2.1.1.2 Mono Conversion Spatial information, as contained in an audio file with more than one channel, is not deemed necessary in the presented work. For this reason, all audio signals are converted to mono by taking the average of all channels.

2.1.1.3 Frame-Based Feature Extraction Rather than performing feature extraction on the entirety of the audio signal, it is common practice to

divide the signal into smaller chunks or *frames*, typically consisting of some 2^n samples (512, 1024, 2048 are often found values). The resulting feature values for each frame form a trajectory of the feature's evolution over time, which can either be used as such or can be averaged. In this work, audio signals are divided into frames with a length of 512 samples. In order to avoid computational errors (such as Not a Number (NaN) in JavaScript) during potentially silent portions of the audio signal, frames with $v_{RMS} < -60\,\text{dBFS}$ (see Root Mean Square (RMS) definition below) are omitted from the feature extraction. The following equations define each feature for a single frame.

2.1.2 Time Domain Features

Time domain features are features derived directly from the discrete-time signal $x[n]$.

2.1.2.1 Duration The overall duration of the signal $x[n]$ in seconds:

$$v_{DUR} = \frac{n}{fs} s, \tag{2}$$

where n is the number of samples and fs is the sampling rate.

2.1.2.2 Root Mean Square (RMS) measures the power of a signal (Lerch, 2012, p.73f). It describes sound intensity and is sometimes used as a simple measure for loudness (Rawlinson et al., 2019a) that does not take the nonlinearity of human hearing into account (Fletcher and Munson, 1933). It is calculated for an audio frame $x[n]$ consisting of n samples such that

$$v_{RMS} = \sqrt{\frac{\sum_{i=1}^{n} x(i)^2}{n}}. \tag{3}$$

2.1.2.3 Zero-Crossing Rate (ZCR) represents the rate of the number of sign changes in a signal. It can be used as a measure of the tonalness of a sound (Lykartsis, 2014) and as a simple pitch detection method for monophonic signals (de la Cuadra, 2019). It is defined as

$$v_{ZCR} = \frac{1}{2 \cdot n} \sum_{i=1}^{n} |sgn[x(i)] - sgn[x(i-1)]|. \tag{4}$$

2.1.3 Frequency Domain Features

Frequency domain features or *spectral* features are derived from the discrete complex spectrum $X(k)$, where k refers to the frequency bin number. $X(k)$ is calculated from $x[n]$ by performing an Fast Fourier Transform (FFT) (for information on the Fourier transform, refer to any signal processing textbook, such as Oppenheim and Schafer (2014)).

2.1.3.1 Spectral Centroid is a measure of the center of gravity of a spectrum. A higher value indicates a brighter, sharper sound (Lerch, 2012). The spectral centroid is defined as

$$v_{SC} = \frac{\sum_{k=0}^{N_{FFT}/2-1} k \cdot |X(k)]^2}{\sum_{k=0}^{N_{FFT}/2-1} |X(k)]^2}. \tag{5}$$

2.1.3.2 Spectral Flatness is a measure for the tonality or noisiness of a signal, defined as the ratio of the geometric and arithmetic means of its magnitude spectrum. Higher values indicate a flatter (and therefore noisier) spectrum, whereas lower values point towards more tonal spectral content. It is defined as

$$v_{SFL} = \frac{\sqrt[N_{FFT}/2]{\prod_{k=0}^{N_{FFT}/2-1} |X(k)|}}{(2/N_{FFT}) \cdot \sum_{k=0}^{N_{FFT}/2-1} |X(k)|}. \tag{6}$$

2.1.3.3 Spectral Kurtosis indicates whether a given magnitude spectrum's distribution is similar to a Gaussian distribution. Negative values result from a flatter distribution, whereas positive values indicate a peakier distribution. A Gaussian distribution would result in a value of 0. Spectral Kurtosis is defined as

$$v_{SKU} = \frac{2 \sum_{k=0}^{N_{FFT}/2-1} (|X(k)| - \mu_{|X|})^4}{N_{FFT} \cdot \sigma_{|X|}^4} - 3, \tag{7}$$

where $\mu_{|X|}$ represents the mean and $\sigma_{|X|}$ the standard deviation of the magnitude spectrum $|X|$.

2.1.3.4 Spectral Skewness assesses the symmetry of a magnitude spectrum distribution. It is defined as

$$v_{SSK} = \frac{2 \sum_{k=0}^{N_{FFT}/2-1} (|X(k)| - \mu_{|X|})^3}{N_{FFT} \cdot \sigma_{|X|}^3}. \tag{8}$$

2.1.3.5 Spectral Slope represents a measure of how sloped or inclined a given spectral distribution is. The spectral slope is calculated using a linear regression of the magnitude spectrum such that

$$v_{SSL} = \frac{\sum_{k=0}^{N_{FFT}/2-1} (k - \mu_k)(|X(k)| - \mu_{|X|})}{\sum_{k=0}^{N_{FFT}/2-1} (k - \mu_k)^2}. \tag{9}$$

2.1.3.6 Spectral Spread is a descriptor of the concentration of a magnitude spectrum around the Spectral Centroid and assesses the corresponding signal's bandwidth. It is defined as

$$v_{SSP} = \frac{\sum_{k=0}^{N_{FFT}/2-1} (k - v_{SC})^2 \cdot |X(k)|^2}{\sum_{k=0}^{N_{FFT}/2-1} |X(k)|^2}. \tag{10}$$

2.1.3.7 Spectral Rolloff measures the bandwidth of a given signal by calculating that frequency bin below which lie κ percent of the sum of magnitudes of $X(k)$. Common values for κ are 0.85, 0.95 (Lerch, 2012) or 0.99 (Rawlinson et al., 2019a). It is defined as

$$v_{SR} = i \Big|_{\sum_{k=0}^{i} |X(k)| = \kappa \cdot \sum_{k=0}^{N_{FFT}/2-1} |X(k)|}. \tag{11}$$

2.1.4 Perceptual Features

Both the time and frequency domain features introduced above are derived from raw audio samples without taking into account any concept of human sound perception. Perceptual features incorporate some sort of model that approximates this perception. While only a single perceptual feature is used in this work, more do exist (see Peeters (2004) for a list of some of them).

2.1.4.1 Total Loudness represents an algorithmic approximation of the human perception of a signal's loudness based on Moore et al. (1997), which uses the Bark scale as introduced by Zwicker (1961). The Total Loudness is the sum of all 24 bands' specific loudness coefficients, defined by Peeters (2004) as

$$v_{TL} = \sum_{i=1}^{24} v_{SL}(i), \tag{12}$$

where

$$v_{SL}(i) = E(i)^{0.23} \tag{13}$$

is the specific loudness of each Bark band (see Moore et al. (1997) for further details).

2.2 Self-Organizing Map

The *self-organizing map* (SOM) is a machine learning algorithm for dimensionality reduction, visualization and analysis of higher-dimensional data. Sometimes also referred to as *Kohonen map* or *network*, it was introduced in 1981 by Teuvo Kohonen (Kohonen, 1990).
The SOM is a variant of an *artificial neural network* that uses an unsupervised, competitive learning process to map a set of higher-dimensional observations (the *input vectors*) onto a regular, often two-dimensional grid or *map* of *neurons* or *nodes* that is easy to visualize. The SOM can be regarded as a nonlinear generalization of a principal component analysis (PCA) (Yin, 2007) or as a quantization of the input data, with the nodes along the map functioning as pointers into that higher-dimensional space. Each node has a position on the lower-dimensional grid as well as an associated position in the input space, which takes the form of a n-dimensional weight vector $m = [m_1, ..., m_n]$, where n is the number of dimensions of the input vectors. Nodes that are in close proximity to each other on the SOM will also have similar weight vectors (Vesanto et al., 2000), although the inverse (neighboring positions in the input space also mapping to neighboring nodes) is not necessarily true (Bauer et al., 1996).
For an in-depth look at the algorithm, its variants and applications, as well as an extensive survey of research on SOMs, the avid reader is referred to Kohonen (2001).

2.2.1 Algorithm Definition

The following definition is based on Kohonen (1990), Kohonen (2005), Kohonen and Honkela (2007) and Bauer et al. (1996).

Consider a space of input data in the form of n-dimensional vectors $x \in \mathbb{R}^n$ and an ordered set of nodes or model vectors $m_i \in \mathbb{R}^n$. A vector $x(t)$ is mapped to that node m_c with the shortest Euclidean distance from it:

$$||x(t) - m_c|| \leq ||x(t) - m_i|| \; \forall i. \tag{14}$$

This "winning" node m_c is referred to as the Best Matching Unit (BMU) for $x(t)$.

During the learning or adaptation phase of the algorithm, all nodes m_i are adjusted by a recursive regression process

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t)), \tag{15}$$

where t is the index of the current regression step, $x(t)$ is an input vector chosen randomly from the input data at this step, c is the index of the BMU for the current input vector $x(t)$ according to equation 14 and $h_{c(x),i}$ represents a so-called *neighborhood function*. The name-giving *neighborhood* is a subset N_c of nodes centered on m_c. At each learning step t, those nodes that are within N_c will be adjusted, whereas those outside of it will not. The reason for employing such a neighborhood function is so that the nodes "doing the learning are not affected independently of each other" (Kohonen, 1990, p.1467) and "the topography of the map is ensured" (Bauer et al., 1996, p.5). At its most basic, the neighborhood function is a decreasing distance function between neurons m_i and m_c. Its most common form, which is also employed in this work, is that of a Gaussian function with its peak at m_c such that

$$h_{c(x),i} = \alpha(t) \exp\left(- \frac{||r_i - r_c||^2}{2\sigma^2(t)} \right). \tag{16}$$

Here, α denotes a learning rate factor or adaptation "gain control" $0 < \alpha(t) < 1$, which decreases over the course of the regression, $r_i \in \mathbb{R}^2$ and $r_c \in \mathbb{R}^2$ are the locations of m_i and m_c on the SOM grid (the lower-dimensional output map, not the input space!), and $\sigma(t)$ is the width of the neighborhood function, which again decreases as the regression step index increases.

2.2.2 Node Initialization

Because of the iterative nature of the SOM algorithm, its outcome depends on the initial positions chosen for the nodes. The method implemented in this work uses random initialization, meaning the starting positions of the nodes are chosen randomly from within the bounds of the input space. An often

employed alternative approach is to first perform a Principal Component Analysis (PCA) on the input data, select the largest d components, where d is the number of desired output dimensions for the SOM, and then distribute the nodes at equidistant intervals along those component vectors.

2.2.3 Input Data Scaling

Some consideration should be given to the dynamic range of the input data across its different dimensions. Are the dimensional ranges comparable in their limits? What about their variance? There does not appear to exist a clear consensus across the literature on whether or not normalization of input data is strictly necessary (Vesanto et al. (2000, p.34), Kohonen (1990, p.1470), Kohonen and Honkela (2007)).

Because the range of the data derived from the audio feature analysis used in this work varies considerably between features, the data for feature n is rescaled to have unit variance by dividing by the features' standard deviation σ_n:

$$x_n = \frac{x_n}{\sigma_n}. \tag{17}$$

2.2.4 Alternative Learning Rate Factors

2.2.4.1 Linear The traditional SOM algorithm uses a learning rate factor α that decreases linearly as a function of the regression step t:

$$\alpha(t) = \alpha_0\left(\frac{1-t}{T}\right), \tag{18}$$

where α_0 is the initially chosen learning rate and T is the total training length or number of regression steps.

Two other approaches to the decreasing learning rate factor were implemented in this work:

2.2.4.2 Inverse The first is a reciprocally decreasing function where

$$\alpha(t) = \frac{\alpha_0}{\left(\frac{1+100t}{T}\right)}. \tag{19}$$

2.2.4.3 BDH The second alternative approach is that of an adaptive local learning rate as developed by Bauer et al. (1996) (BDH algorithm, also see Merenyi et al. (2007)):

$$\alpha(t) = \alpha_0\left(\frac{1}{\Delta t_c}\left(\frac{1}{|x(t) - m_c|^n}\right)\right)^m, \tag{20}$$

where Δt_c represents the time since the current BMU for the current input vector was last selected as a BMU for any vector and m is a newly introduced, free control parameter. For a more complete review of the uses of this algorithm, the reader is referred to the original paper (Bauer et al., 1996) as well as Merenyi et al. (2007).

2.2.5 Forced Node Population

One of the aspects that make the SOM immediately interesting for music technology applications is that it is fundamentally based on a regular grid structure, which opens a direct connection to the grid as a musical structure.

In order to avoid "empty" nodes on the SOM - meaning nodes to which no input vectors are mapped - a post-processing extension was added to the original algorithm that inverts the mapping process, explicitly iterates over empty nodes and assigns each one that input vector which is closest. This algorithm extension, which we call Forced Node Population (FNP), executes the following sequence after the regular SOM has been calculated:

1. Select a random empty node.
2. Find closest vector for that node and assign it to this node.
3. Remove this vector from the possible choices.
4. Repeat.

This process is only performed once for all nodes that are empty immediately after the initial SOM calculation. It is possible that the forced node population creates new empty nodes, but in order to minimize distortion introduced by this procedure (see Results in section 5), it is not repeated. For a look at the results of this algorithm extension, please refer to Results (section 5).

3 Implementation

This is the Implementation.

3.1 Groundwork: CataRT Extension

3.2 SOM Browser

4 Evaluation

This is the Evaluation.

4.1 Sound Corpus Selection

A crucial aspect for the evaluation of the work presented in this thesis is the choice of an appropriate data set of audio files to serve as a prototypical sound corpus. Ideally, two key conditions should be met by this corpus. It should be *ecologically valid*, meaning here that it should approximate a real-world sample library that would actually be used by contemporary music producers, and it should be a *well-established* data set which has been validated through use in other research, allowing for direct comparisons between results. Preferably, something akin to the Giant Steps data sets (Knees et al., 2015) for tempo and key detection should be used. In addition to identifying the aforementioned two conditions, the decision was made to only select "one-shot" drum and percussion sounds (meaning single instrument hits, no loops or other longer sounds) in order to evaluate a single, concrete use case and limit the scope of this evaluation.

Data sets used in previous research vary and it is often not possible to clearly establish provenance due to insufficient information being given by the authors (see for example Fried et al. (2014) and Shier et al. (2017), two papers which present important related work but fail to clearly identify the source of their employed sound files). Two established databases that have been cited in the literature are ENST-Drums (Gillet and Richard, 2006) and the RWC Music Database (Goto et al., 2002). However, neither of these data sets proved appropriate for this evaluation since they both contain only acoustic source material and, especially in the case of RWC, largely consist of longer musical passages instead of the single "one-shot" hits mentioned above.

For the reasons outlined above, the author decided to forego the condition that the selected sound corpus be a data set well-established through previous research. Because of this, more emphasis is placed on the requirement for ecological validity. In order to maximize real-world conditions, the sample library *Drum Essentials* (Ableton, 2019b) was selected to serve as a sound corpus for this evaluation. It is a collection of samples created by the German music software company Ableton AG that is distributed to owners of the company's flagship product, the Digital Audio Workstation (DAW) *Ableton Live* (Ableton, 2019a). As part of a commercially available product, this corpus of sound files does not just approximate a real-world sample library, it is an actual example of such a library and is, for the purpose of this

thesis, considered representative of sample libraries used in a modern music production workflow. One additional benefit of using the selected sample library is the advantage of a single, clearly identifiable source of the data - it is made available as a professional product by Ableton AG. An alternative approach would have been to manually select sounds from places like *Freesound.org* (Font et al., 2013), where all files are licensed in a way that makes them free to use, but their quality is not guaranteed to be consistent, or to scour through sample libraries shared on various online forums, which brings along issues of copyright and expired links, making it hard to trace the files' origins.

The *Drum Essentials* collection as distributed by Ableton consists of 1181 one-shot samples, each in a separate audio file, as well as supplementary content, such as MIDI clips and effects presets. Only the raw audio files are used in the presented work. These sound files present a mixture of acoustic and electronic sounds stemming from a variety of drums and percussion instruments. The library is organized by instrument group, of which there are 17 in total. The names of these groups, as well as the number of sounds per group can be found in table 1. Some sound files appear in more than one group. These duplicates have been removed, so that every sound only appears once throughout the entire data set. The remaining number of sound files is 1081.

4.2 Metrics for SOM Analysis

4.2.1 SOM-Induced Quantization and Influence of Forced Node Population

4.2.2 Map Emptiness

4.2.3 Vector-Node Ratio

4.3 Online Sound Similarity Survey

Maybe not even do this bit???

4.4 Semistructured User Interviews

In order to evaluate the SOM Browser application prototype presented in this thesis, five semi-structured interviews with working audio professionals were conducted. These interviews were conducted by the author and consisted of a set of questions as well as observed user interaction with the prototype software. For this evaluation, a guide including questions outlining the structure of the interview as well a set of ratings scales was created. Subjects were

Drum Essentials	
Instrument Category	**Count**
Bell	19
Bongo	6
Clap	71
Conga	27
Cymbal	54
Electronic Percussion	49
FX Hit	64
Hihat	167
Kick	166
Misc. Percussion	64
Ride	40
Rim	65
Shaker	39
Snare	181
Tambourine	23
Tom	138
Wood	8

Tab. 1: Sound file counts per instrument category of the *Drum Essentials* sample library

asked about their experience with sample libraries and their current workflow, and to interact with a sample library in a file browser environment as well as using the SOM Browser software. Audio from the conversations was recorded and subsequently analyzed.

4.4.1 Motivation to Conduct Interviews

This evaluation procedure entails two aspects, namely a semi-structured interview series and a qualitative analysis of the collected responses. The decision to conduct qualitative interviews stems from the exploratory nature of the presented work. In order to assess the merit of the developed interface in its present state, direct feedback from potential users was sought, which Lazar et al. (2017) refers to as "fundamental to human-computer-interaction (HCI) research" (see Lazar et al. (2017, p.187)). But the motivation for a direct conversation with users was not only to evaluate the presented interface

proposition, but also for these interviews to serve as an exploration of users' current situation, to hear about their own experience of it and to see what advantages and shortcomings they identify in their present workflows. In short, these interviews were motivated by a desire to gain some understanding of the complex situation that is sample library interaction in a music production environment and to gauge initial reactions to the developed prototype alternative. The semi-structured approach was chosen in order to be able to react to interviewees' responses more freely and allow the interviewer to ask follow up questions when deemed necessary. Naturally then, the gathered responses cannot simply be quantified, which makes a qualitative approach to their analysis a fitting choice.

There are of course downsides to the chosen approach. Conducting interviews is time-consuming, as it has to be done on a one-on-one basis and often (as in the case of this work) in person. After the interview is over, additional time and effort goes into transcribing and annotating the responses. This severely limits the number of participants that can feasible be recruited for a study, as is evident by the small number of five participants here. Lazar et al. (2017) identifies another disadvantage of interviews: "[...] data collection that is separated from the task and context under consideration [...] suffer[s] from problems of recall. [...] [I]t is, by definition, one step removed from reality" (Lazar et al., 2017, p.188ff.). Because of this, we follow the authors' suggestion of combining the interview with user observation.

4.4.2 Interview Subject Selection

The SOM Browser application is not aimed at the general population. Instead, it has been designed for specialized users that work in modern music production, as they constitute the potential future user base of an application like the one presented here.

In order to increase the validity and relevance of potential subjects' responses, the decision was made to interview only working professionals for this evaluation and to not include hobbyists or people without any experience in music production.

Subjects were recruited by inquiring about qualified candidates (in other words, people working professionally in modern music production) in the wider circle of acquaintances of the author. No compensation was offered and only sparse information about the nature of the research was given beforehand in order to minimize the possibility of instilling biases in subjects. Most importantly, subjects were asked to participate in an interview about sample library organization, but were not told that they would be shown

software developed by the author.

4.4.3 Informed Consent Form

For the purpose of documenting participants agreement to be interviewed, an informed consent form was created for the interview series. This document outlines basic information about the purpose and content of the interview and its duration. It also lists all data that will be collected and explains the procedure used for data anonymization in order to protect subjects' privacy. Lastly, it informs participants of their rights to withdraw their consent to the usage of their data for research purposes and have it erased. This form was based on a template provided by the ethics commission of Technische Universität Berlin (TU Berlin) on their website (TU Berlin, 2019). The form used by the author can be found in XXX REF APPENDIX HERE XXX.

4.4.4 Test Subject Code Design

To ensure proper data anonymization, a test subject code was used. This code is comprised of a series of letters and numbers and was created at the beginning of the interview by the subjects themselves according to a set of instructions. All data and responses of the subjects were directly labelled with this code, so that individuals' names were never used. This code design procedure was again based on a template by the ethics commission of TU Berlin and can be found on the same website as the information concerning consent forms (TU Berlin, 2019). The instruction sheet that was distributed to subjects can be found in XXX REF APPENDIX HERE XXX.

4.4.5 Interview Structure

The guide developed for this interview can be found in XXX REF APPENDIX HERE XXX It outlines a three part structure: first, some general questions about subjects' usage of sample libraries. Second, some guided interaction with a predetermined sample library in a traditional file browser structure on a computer. In the third section, the SOM Browser application is finally introduced and subjects are asked to use it and describe their impression of it.

4.4.6 Question Design

The general composition employed for most questions is twofold, combining closed- and open-ended approaches: first, participants are asked to give a rating on a predefined scale (see 4.4.7 below). Then, participants are free to

elaborate on their answer and explain their rating. If they don't initiate this themselves, a follow-up question along the lines of "Could you tell me why you chose this rating?" is asked.

4.4.7 Selection of Ratings Scales

In order to record subjects' ratings, 6 point Likert scales were used (as is common in Human-Computer Interaction (HCI) research, see Lazar et al. (2017, p.31, p.93)). The difference between even and uneven anchor counts in Likert scales lies in the presence (in the case of uneven anchor counts) or lack (for even counts) of a "neutral" middle option. Choosing scales without neutral mid-points was motivated by a desire to encourage subjects to make a definite choice with regard to their rating. For a short look at the effects of eliminating the mid-point, see Garland (1991). The scales presented to subjects were explicitly labeled textually instead of numerically. The anchor points were designed using two polar adjectives (such as "positive" and "negative") and a consistent, three-tiered set of adjective qualification with "very" marking the strongest option, followed by the adjective without qualifier and then "somewhat" as the weakest variant. The resulting scale for a positive/negative rating is composed of the following anchors: very positive, positive, somewhat positive, somewhat negative, negative, very negative. The selection of these qualifiers and appropriate anchors in general was inspired partially by Vagias (2006). The full set of scales used for the conducted interviews can be found in XXX REF APPENDIX HERE XXX.

4.4.8 Questions Used

In section 1, which serves as an introduction for the interviewee, general administrative requirements such as the signing of the consent form and a topical introduction of the research are taken care off. This is then followed by two simple Yes/No questions to establish whether the subject works with third-party and personally created sample libraries (see questions 1.1 and 1.2).

Section 2 begins with a presentation of the *Drum Essentials* sample library to the subject. This presentation includes the information that it is a library of drum samples that consists of around 1000 sound files which are organized in subfolders according to the respective instrument, such as kick drum, snare drum, hi-hat, and so forth. The interviewee is invited to explore the sample library using the laptop that it is being presented on.

Then, in question 2.1, subjects are asked to describe how to approach familiarizing themselves with the provided sample library in order to use its contents in a hypothetical work project of theirs.

Question 2.2 follows this up with a request for a rating of the subject's level of satisfaction with the workflow that they outlined.

In the third and final section of the interview, the SOM Browser software is introduced to participants. At first, a general overview of the interface is given, in which the interviewer mentions the map layout in the middle (without explaining the nature of its organization), the file list on the left, the file info panel on the right and the favorites bar at the bottom.

The subject is then asked to try out the software and explore its interface for a short period of time. Thereafter, they are asked to give a rating of their overall first impression of the software on a positive/negative scale (see question 3.1). Then, a follow-up question about their opinion on what does or does not work is posed.

In question 3.2, subjects are required to rate the interface's ease of use.

Question 3.3 inquires specifically about the understandability of the language used.

3.4 and 3.5 are open-ended questions aimed at subjects' interpretation of the organization of sounds in the map layout: 3.4. asks what subjects think about the organization, while 3.5 inquires specifically about a guess as to what the axes represent.

Question 3.6 then asks subjects to state whether or not they have a preference between the traditional file browser layout presented in section 2 or the SOM Browser interface shown in section 3.

The last ratings question of the interview, 3.7 requests interviewees to assess their level of comfortability with the software.

Finally, in 3.8 subjects are asked if they would consider using the presented software tool and what changes they would like to see.

The full interview guide including all questions can be found in XXX REF APPENDIX HERE XXX.

5 Results

This is the Results section.

6 Discussion

This is the Discussion.

6.1 Outlook

7 References

Ableton, AG (2019a): *Ableton Live 10*. Online. URL `https://www.ableton.com/en/live/`. Access 7.2.2019.

Ableton, AG (2019b): *Drum Essentials*. Online. URL `https://www.ableton.com/en/packs/drum-essentials/`. Access 7.2.2019.

Bauer, H.-U.; Ralf Der; and Michael Herrmann (1996): "Controlling the magnification factor of self-organizing feature maps." In: *Neural computation*, **8**(4), pp. 757–771.

de la Cuadra, Patricio (2019): "Pitch Detection Methods Review." URL `https://ccrma.stanford.edu/~pdelac/154/m154paper.htm`.

DeSieno, Duane (1988): "Adding a conscience to competitive learning." In: *IEEE international conference on neural networks*, vol. 1. Institute of Electrical and Electronics Engineers New York, pp. 117–124.

Fasciani, Stefano (2016): "TSAM: a tool for analyzing, modeling, and mapping the timbre of sound synthesizers." In: .

Fiebrink, Rebecca and Baptiste Caramiaux (2016): "The machine learning algorithm as creative musical tool." In: *Handbook of Algorithmic Music*.

Fletcher, Harvey and Wilden A Munson (1933): "Loudness, its definition, measurement and calculation." In: *Bell System Technical Journal*, **12**(4), pp. 377–430.

Font, Frederic; Gerard Roma; and Xavier Serra (2013): "Freesound technical demo." In: *Proceedings of the 21st ACM international conference on Multimedia*. ACM, pp. 411–412.

Fried, Ohad; Zeyu Jin; Reid Oda; and Adam Finkelstein (2014): "AudioQuilt: 2D Arrangements of Audio Samples using Metric Learning and Kernelized Sorting." In: *NIME*. pp. 281–286.

Garland, Ron (1991): "The mid-point on a rating scale: Is it desirable." In: *Marketing bulletin*, **2**(1), pp. 66–70.

Gillet, Olivier and Gaël Richard (2006): "ENST-Drums: an extensive audio-visual database for drum signals processing." In: *ISMIR*. pp. 156–159.

Goto, Masataka; Hiroki Hashiguchi; Takuichi Nishimura; and Ryuichi Oka (2002): "RWC Music Database: Popular, Classical and Jazz Music Databases." In: *ISMIR*, vol. 2. pp. 287–288.

Knees, Peter; et al. (2015): "Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections." In: *ISMIR*. pp. 364–370.

Kohonen, Teuvo (1990): "The Self-Organizing Map." In: *Proceedings of the IEEE*, **78**(9), pp. 1464–1480.

Kohonen, Teuvo (1997): "Exploration of very large databases by self-organizing maps." In: *Proceedings of International Conference on Neural Networks (ICNN'97)*, vol. 1. IEEE, pp. PL1–PL6.

Kohonen, Teuvo (2001): *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Heidelberg: Springer.

Kohonen, Teuvo (2005): "The Self-Organizing Map (SOM)." URL `http://www.cis.hut.fi/somtoolbox/theory/somalgorithm.shtml`.

Kohonen, Teuvo and Timo Honkela (2007): "Kohonen network." URL `http://www.scholarpedia.org/article/Kohonen_network`.

Lazar, Jonathan; Jinjuan Heidi Feng; and Harry Hochheiser (2017): *Research methods in human-computer interaction*. Morgan Kaufmann.

Lerch, Alexander (2012): *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press.

Lykartsis, Athanasios (2014): *Evaluation of accent-based rhythmic descriptors for genre classification of musical signals*. Master's thesis, Master's thesis, Audio Communication Group, Technische Universität Berlin ....

Mathieu, Benoit; Slim Essid; Thomas Fillon; Jacques Prado; and Gaël Richard (2010): "YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software." In: *ISMIR*. pp. 441–446.

Mathieu, Benoit; Slim Essid; Thomas Fillon; Jacques Prado; and Gaël Richard (2019): "Yaafe - audio features extraction." URL `http://yaafe.sourceforge.net/`.

Mayring, Philipp (2010): "Qualitative Inhaltsanalyse." In: *Handbuch qualitative Forschung in der Psychologie*. Springer, pp. 601–613.

Merenyi, Erzsbet; Abha Jain; and Thomas Villmann (2007): "Explicit magnification control of self-organizing maps for "forbidden" data." In: *IEEE Transactions on Neural Networks*, **18**(3), pp. 786–797.

Moffat, David; David Ronan; Joshua D Reiss; et al. (2015): "An evaluation of audio feature extraction toolboxes." In: .

Moore, Brian CJ; Brian R Glasberg; and Thomas Baer (1997): "A model for the prediction of thresholds, loudness, and partial loudness." In: *Journal of the Audio Engineering Society*, **45**(4), pp. 224–240.

Oppenheim, Alan V and Ronald W Schafer (2014): *Discrete-time signal processing*. Pearson Education.

Peeters, Geoffroy (2004): *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. Tech. rep., IRCAM.

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2015): "Meyda: an audio feature extraction library for the web audio api." In: *The 1st Web Audio Conference (WAC). Paris, Fr.*

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2019a): "Meyda: Audio feature extraction for JavaScript." URL `https://meyda.js.org/audio-features`.

Rawlinson, Hugh; Nevo Segal; and Jakub Fiala (2019b): "Meyda: Audio feature extraction for JavaScript." URL `https://github.com/meyda/meyda`.

Scholler, Simon and Hendrik Purwins (2010): "Sparse coding for drum sound classification and its use as a similarity measure." In: *Proceedings of 3rd international workshop on Machine learning and music*. ACM, pp. 9–12.

Shier, Jordie; Kirk McNally; and George Tzanetakis (2017): "Analysis of Drum Machine Kick and Snare Sounds." In: *Audio Engineering Society Convention 143*. Audio Engineering Society.

TU Berlin, Ethik-Kommission (2019): *Ethik-Kommission*. URL `https://www.ipa.tu-berlin.de/menue/einrichtungen/gremienkommissionen/ethik_kommission/`.

Vagias, Wade M (2006): "Likert-type Scale Response Anchors. Clemson International Institute for Tourism." In: *& Research Development, Department of Parks, Recreation and Tourism Management, Clemson University.*

Vesanto, Juha; Johan Himberg; Esa Alhoniemi; and Juha Parhankangas (2000): "SOM toolbox for Matlab 5." In: *Helsinki University of Technology, Finland*, p. 109.

Villmann, Thomas and Jens Christian Claussen (2006): "Magnification control in self-organizing maps and neural gas." In: *Neural Computation*, **18**(2), pp. 446–469.

Yin, Hujun (2007): "Nonlinear dimensionality reduction and data visualization: a review." In: *International Journal of Automation and Computing*, **4**(3), pp. 294–303.

Zwicker, Eberhard (1961): "Subdivision of the audible frequency range into critical bands (Frequenzgruppen)." In: *The Journal of the Acoustical Society of America*, **33**(2), pp. 248–248.

Appendices

A LaTeX Sources

The LaTeX sources for this work can be found in XXX.

B Thesis Bibliography

The references used in this work can be found in XXX.

Acronyms

BMU Best Matching Unit.

DAW Digital Audio Workstation.

FFT Fast Fourier Transform.

FNP Forced Node Population.

HCI Human-Computer Interaction.

NaN Not a Number.

PCA Principal Component Analysis.

SOM Self-Organizing Map.

TU Berlin Technische Universität Berlin.

List of Figures

List of Listings

List of Tables

Digital Resource

This page holds a data disk.