Tulilaulu, Aurora; Nelimarkka, Matti; Paalasmaa, Joonas; Johnson, Daniel; Ventura, Dan; Myllys, Petri; Toivonen, Hannu

Data Musicalization

Published in:
ACM Transactions on Multimedia Computing, Communications and Applications

DOI:
10.1145/3184742

Published: 01/04/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Tulilaulu, A., Nelimarkka, M., Paalasmaa, J., Johnson, D., Ventura, D., Myllys, P., & Toivonen, H. (2018). Data Musicalization. *ACM Transactions on Multimedia Computing, Communications and Applications*, *14*(2), 1-27. Article 47. https://doi.org/10.1145/3184742

Data Musicalization

Aurora Tulilaulu, University of Helsinki
Matti Nelimarkka, Aalto University & University of Helsinki
Joonas Paalasmaa, Beddit Ltd.
Daniel Johnson, Brigham Young University
Dan Ventura, Brigham Young University
Petri Myllys, University of Helsinki
Hannu Toivonen[1], University of Helsinki

Data musicalization is the process of automatically composing music based on given data, as an approach to perceptualizing information artistically. The aim of data musicalization is to evoke subjective experiences in relation to the information, rather than merely to convey unemotional information objectively. This paper is written as a tutorial for readers interested in data musicalization. We start by providing a systematic characterization of musicalization approaches, based on their inputs, methods and outputs. We then illustrate data musicalization techniques with examples from several applications: one that perceptualizes physical sleep data as music, several that artistically compose music inspired by the sleep data, one that musicalizes on-line chat conversations to provide a perceptualization of liveliness of a discussion, and one that uses musicalization in a game-like mobile application that allows its users to produce music. We additionally provide a number of electronic samples of music produced by the different musicalization applications.

CCS Concepts: •**Information systems** → **Multimedia information systems; Multimedia content creation;** •**Human-centered computing** → **Sound-based input / output; Auditory feedback;** •**Applied computing** → **Sound and music computing;** •**Computing methodologies** → *Artificial intelligence; Machine learning;*

Additional Key Words and Phrases: Data Musicalization; Sonification; Automated Composition; Music; Data Analysis; Computational Creativity

ACM Reference Format:
Aurora Tulilaulu, Matti Nelimarkka, Joonas Paalasmaa, Daniel Johnson, Dan Ventura, Petri Myllys and Hannu Toivonen, 2016. Data Musicalization. *ACM Trans. Multimedia Comput. Commun. Appl.* 0, 0, Article 0 (2016), 22 pages.
DOI: 0000000.0000001

1. INTRODUCTION

Data Musicalization is the process of using automatic composition methods to create music from given data. Data musicalization can have two subtly different purposes:

— To perceptualize data, e.g., to facilitate data understanding and analysis. Perceptualization of data as music is motivated by the current multitude of data sources that have also lead to an increasing demand for information visualization.
— To create an artistic artefact from data with an intrinsic aesthetic value. This is interesting in its own right, but automated creation of artistic musical artefacts from data can somewhat paradoxically also serve the purpose of self-expression: if the data is personal to the user, the user can enjoy a feeling of creativity because the unique music originates from their data.

Data musicalization is closely related to *sonification* (e.g., [Bonebright and Flowers 2011; Hansen and Rubin 2001]). Sonification and data musicalization share the first function above, to perceptualize data as

[1]Corresponding author

This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland (Grant 118653) and by the European Commission (FET grant 611733, ConCreTe). This work has also been supported by the Finnish Funding Agency for Innovation, through LEAD-project.
Author's addresses; Aurora Tulilaulu and Petri Myllys and Hannu Toivonen, Department of Computer Science and Helsinki Institute for Information Technology HIIT, University of Helsinki; Matti Nelimarkka, Department of Computer Science and Helsinki Institute for Information Technology HIIT, Aalto University; Joonas Paalasmaa, Beddit Ltd., Finland; Daniel Johnson and Dan Ventura, Computer Science Department, Brigham Young University.

DOI: 0000000.0000001

audio. There are two major differences in how they address this goal, however. First, data musicalization is based on intentionally composing music, ultimately in a creative process, while sonification typically uses relatively direct mappings from data to audio and there is no guarantee of the musicality or artistic quality of the results. Second, data musicalization aims for subjective and even emotional experiences while sonification traditionally focuses more on conveying neutral, non-affective information.

Consider, for instance, following a topic on Twitter in the background while working on something else. A suitable musicalization of the stream of tweets would leave your eyes and mind free for the actual work while still giving an audible impression of characteristics such as the rate of tweets or their valence. A simple sonification would likely be non-musical, while musicalization aims specifically to be musical and thus more pleasant or interesting, potentially avoiding issues like user fatigue and attenuation.

In contrast to the perceptualization of data, providing aesthetic pleasure or the joy of creativity is a more challenging but perhaps also more interesting goal of data musicalization. Can data be turned into music that gives pleasure to the owner of the data? Could the user even feel the joy of creation, especially if they produced the data and thus indirectly also the music? An example application is musicalization of sleep measurements: a novel piece of music is inspired by biophysical measurements of one's sleep during the night. At its best, the experience could be highly personal and result in a unique form of self reflection.

This paper proposes data musicalization as a topic of its own identity. We characterize and position it as a research field, and we provide concepts to analyze and describe data musicalization applications. The body of this article consists of descriptions of data musicalization applications, providing practical material and examples to help interested readers get started with data musicalization. The structure of the paper is the following.

In Section 2 we suggest a small set of defining characteristics of data musicalization using concepts from better established related fields. Using these characteristics, we compare and contrast data musicalization to its closest relative, sonification. We then provide distinctive characteristics for analyzing and comparing different data musicalization applications.

In Section 3 we outline an example process of data musicalization, including both musical design, automated composition and engineering tasks.

We believe that the concept of data musicalization is best explained by concrete applications. We thus proceed to describe several example applications of data musicalization in different domains: sleep analysis (Section 4), online chat monitoring (Section 5) and mobile music production (Section 6). We provide sample music files produced by the methods described in this article, so readers can listen to the results (see Online Appendix). We have also made some of the implementations available as source code, for those interested in developing their own applications or carrying out research in data musicalization (see links in Supplementary Materials).

Finally, Section 7 contains the discussion and conclusions.

2. DEFINING AND CHARACTERIZING DATA MUSICALIZATION

The term 'data musicalization' was first suggested by Tulilaulu et al. [2012] in the context of automated composition of music from sleep measurements. The field thus is young and there is essentially no literature on 'data musicalization'. The idea of turning data into sound is very old, however, and has been researched under sonification for many years. In this section, we will first position data musicalization as a research field, especially with respect to sonification. We will then provide and discuss characteristics that are distinctive for different data musicalization applications.

2.1. Characteristics of Data Musicalization as a Field

Data musicalization is located in the intersection of sonification and automatic composition of music. While the relationship of data musicalization to automatic composition is relatively clear — data musicalization uses automatic composition methods — it is perhaps less obvious how data musicalization differs from sonification.

It is typical for data musicalization to carry out the creative and open task of composing novel music. Unlike sonification, it therefore extends to the fields of automated composition [Roads 1996; Fernández and Vico 2013] and computational creativity [Cardoso et al. 2009; Colton and Wiggins 2012; Papadopoulos and Wiggins 1999]. In contrast, sonification software typically implements a much more direct mapping between input data and audio.

When exactly do we talk about musicalization (and not sonification)? What are its defining characteristics? An analysis of concepts in the fields of sonification [Walker and Nees 2011], ambient information systems

Table I. Shared characteristics of data musicalization; compared to sonification for contrast

	Aim	Output vs. Input			Operation		
	Aim	Expressions[1]	Symbolicness[2]	Information capability[2,3]	Method to generate sound[3,4]	Amount of musical knowledge[4]	Aesthetic measure[1,3]
Musicalization	Create a musical experience from data	Music	Metaphorical	Low	Composition of music inspired by data	Medium to high	Implicit or explicit
Sonification	Convey information using sound	Non-speech audio	Direct to metaphorical	Low to high	Fixed or interactive mapping from data	None to low	Implicit

[1][Colton et al. 2011], [2][Pousman and Stasko 2006], [3][Walker and Nees 2011], [4][Roads 1996]

[Pousman and Stasko 2006] and computational creativity [Colton et al. 2011] have lead us to seven key characteristics of data musicalization (cf. columns of Table I). We divide these salient characteristics into three broad classes: *aim* of the system, relationship between *output and input* of the system, and *operation* of the system.

We next present these characteristics and use them to define data musicalization (Table I). For illustration and contrast, we also describe sonification using the same characteristics.

Aim:
— The primary *aim* of musicalization is to produce a musical experience; the primary aim of sonification is to convey information. This difference in their emphases is reflected in their other characteristics below.

Output vs. Input:
— The *expressions* [Colton et al. 2011] (i.e., output) created by musicalization and sonification are music and non-speech audio, respectively. While the broad category of non-speech audio includes music, the output of sonification is rarely characterized as music.
— *Symbolicness* and *representation fidelity* refer to representational capabilities of the output and relate to its semiotics [Pousman and Stasko 2006; Walker and Nees 2011]. The relationship between the audio representation and the perceptualized phenomenon can be, for example, direct (the sound refers to the phenomenon), ecological (the sound is related to phenomena) or metaphorical (the link between sound and phenomena is emblematic). Sonification in general can have any of these types of fidelity, depending on the phenomenon, while musicalization is inherently metaphorical in nature.
— *Information capability* refers to the number of input information sources or variables that a sonification method can convey [Pousman and Stasko 2006]. In sonification, the listener's ability to separate input variables and therefore the polarity of sounds is critical [Walker and Nees 2011]. Musicalization, on the other hand, typically affords less (information) representational ability due to its emphasis on the musicality of the result over the amount of information conveyed, as well as due to the inherent limitations of music to convey information.

Operation:
— *Methods to generate sound* from data vary considerably between systems. For traditional sonification, two categories of methods are usually considered: fixed parametric mappings from certain attributes to outcomes, and interactive methods in which the mapping between the data and the output can be changed by the user [Walker and Nees 2011]. In musicalization, on the other hand, the mappings are usually indirect, the musicalization software has more (creative) freedom, and it is musically informed (cf. aesthetic measure).
— The *amount of musical knowledge* that musicalization requires is obviously higher than that needed for sonification. The knowledge can be coded manually but often is also acquired by machine learning from a corpus of existing music [Roads 1996].
— An *aesthetic measure* indicates what the system considers good output [Colton et al. 2011], and it is obviously tightly linked to the aim of the system. In the case of both musicalization and sonification,

the system designer exercises their (implicit) aesthetics at design time to obtain audio results that match their preferences. In interactive systems, where the end user can affect the mappings used and sounds produced, the user's aesthetics are also considered. In data musicalization, the system itself can have aesthetic preferences, either manually coded or obtained by machine learning. Explicit aesthetic measures allow the system to express its (musical) goals and to assess the quality of its possible outputs at run-time and make decisions or adjustments accordingly.

The above analysis shows that musicalization has a distinctive profile when compared to sonification. On one hand, musicalization operates under more stringent constraints, as it is supposed to always produce music, which is an *artistic* goal. On the other hand, musicalization systems have more freedom in relating data to sound since they do not aim to convey information faithfully. They can take special advantage of this freedom when they have explicit, internal aesthetic measures to guide them towards better results.

2.2. Distinctive Characteristics of Data Musicalization Applications

We now move on to define a set of characteristics that can be used to analyze and describe different data musicalization applications, providing conceptual tools for systematization of the field. (This goal differs from that of Table I, which gives characteristics that are typically *shared* by data musicalization applications and thus define data musicalization as a field.)

Data musicalization has a large range of potential applications, as we will illustrate in the sections to follow. We have identified the following seven key dimensions or characteristics along which applications may differ, and which help characterize how data is being musicalized (cf. columns of Table II).

(1) The specific *aim of the system* can vary from art to entertainment and to transfer of information, even if the shared aim of data musicalization systems is to create a musical experience from the input data.
(2) The domain and nature of the *input data*, used by the system to inspire or instruct composition of music, is very much application dependent.
(3) A wide range of *composition methods* are available for translating the input data to music. We do not provide a classification of composition methods here, but the next two characteristics relate to how music is produced.
(4) The *amount of musical knowledge* incorporated in the system affects the musical quality of the output: a higher amount of knowledge helps the system produce music of the desired genre and style. The musical knowledge can be coded in various ways, depending on the composition methods and the methods of obtaining the knowledge.
(5) The *way of obtaining the musical knowledge* often has both manual components as well as automatically trained, machine learning based components.
(6) By *operating mode* we mean a distinction between batch processing (all data is turned into music at the same time), real-time processing (a stream of data is turned into music continuously as new data arrives), and interactive processing (real-time processing where a user generates or controls the data stream).
(7) Different *modalities* can be used, in addition to audio, in applications where data musicalization is only part of the system output. For instance, a system could compose music to complement its textual or visual output.

Data musicalization is new as a concept, so previously published examples barely exist to illustrate the characteristics of Table II. We provide in Sections 4–6 several examples, including an overview of the one where the concept was originally proposed [Tulilaulu et al. 2012]. We next use these and other, related applications to illustrate the distinctive characteristics (Table II).

In Section 4 we describe four applications in sleep musicalization, illustrating several different uses of data musicalization in the same domain. The first application (Section 4.2) produces an (entertaining) perceptualization of sleep patterns during a single night, aiming at using music to make sleep analysis and tracking more fun. It is based on hand-built Markov models that produce music matching the sleep stages and other sleep measurements.

The three subsequent applications compose more artistic music from the same data, aiming less at describing the data and more at producing an interesting piece of music inspired by the data. They are also all characterized by use of machine learning in the composition methods rather than just manually specified rules. The first application of these three translates the sleep signal deterministically to a melody and then uses a hidden Markov model (HMM) to harmonize it; the HMM has been obtained automatically by analysis of harmonizations of chorales (Section 4.3.1). This application has the least amount of musical knowledge or

Table II. Distinctive chracteristics of musicalization applications

Application	Aim	Input data	Composition methods	Amount of musical knowledge	Way of obtaining musical knowledge	Operating mode	Modalities
Sleep musicalization as perceptualization (Section 4.2)	Entertaining transfer of information	Sleep measurements	Stochastic, based on Markov models	High	Hand-built	Batch	Audio
Measurement-inspired sleep musicalization (Section 4.3.1)	Artistic expression	Sleep measurements	Deterministic pitch detection from sleep signal, harmonization using an HMM	Low	HMM trained with chorales, some components hand-built	Batch	Audio
Emotionally targeted sleep musicalization (Section 4.3.2)	Artistic expression	Sleep measurements	Stochastic, based on various models	High	Models trained with emotionally labeled music, some components hand-built	Batch	Audio
Motivic composition using sleep measurements (Section 4.3.3)	Artistic expression	Sleep measurements	Stochastic, using genetic algorithms (GA) for melody and HMM for harmonization	High	GA with hand-built functions, HMM trained with western rhythm music	Batch	Audio
Chat musicalization (Section 5)	Ambient transfer of information	On-line chat	Stochastic, based on Markov models	High	Hand-built	Real-time	Audio (ambient; complements text)
Musicreatures mobile app (Section 6)	Artistic expression	User gestures (that control visual objects)	Deterministic Euclidian rhythms from visual objects	High	Hand-built	Interactive	Audio-visual

constraints at its disposal. The second artistic application maps sleep stages to emotions and then composes music to express the respective emotions (Section 4.3.2). In this application, emotionally labeled training data has been used to obtain the models for composition. The third artistic application extracts initial motifs from the sleep signal and then uses genetic algorithms (GA) to evolve them to melodies (Section 4.3.3). It relies on hand-built fitness functions and musical knowledge within them; harmonization is again based on HMMs trained with existing music.

In Section 5 we describe an application in online chat musicalization. The goal of the system is to produce an ambient perceptualization of the liveliness and affect of online discussions. Unlike the previous applications, this one works in real-time. The application is also not entirely audio, but the music is intended to augment the original text used as input to the application. In this more controlled setting, we also describe a small user-study on chat musicalization, examining the link between musicalization and the online discussion.

Finally, in Section 6, we describe the Musicreatures mobile application that turns the iPhone into a musical instrument. In the app, the user controls visual objects on the screen by tilting and shaking the phone, the objects in turn control the music generated. This setup gives the user control over the music but only via the objects, while the system design ensures that the objects always produce music within a desired range. Musicreatures illustrates use of musicalization in a game-like setting that is both interactive and audio-visual, unlike the other applications.

There have been previous efforts to musicalize data, even if not under that particular name. We next give examples from previous work based on their historical value or similarity to our cases.

An early example of something that could perhaps be called data musicalization is given by Vickers and Alty [1996; 2005]. Their CAITLIN system sonifies Pascal source code by mapping constructs of the program (if, case, repeat, while, and for) to short musical tunes. This is a direct mapping, this time between data and musical fragments, with little room for (creative) composition of music. Similarly, Flowers et al. [2005] discuss the opportunities to develop sonification as part of data analysis packages. Their ideas include auditory scatterplots, which support the display of data. Recently, Tsuchiya et al. [2015] proposed an API (application programming interface) for sonification with musical structure models, allowing one to map data to high-level musical parameters. Such a tool could be useful for building data musicalization applications. Several toolkits have been built also for interactive sonification. For example, LISTEN [Lodha et al. 1996; Wilson and Lodha 1996] and PORSONIFY [Madhyastha and Reed 1995] both allow their users to interactively map input data to outputs like timbre and pitch.

Data has been sonified before with artistic goals. For instance, CLIMATE SYMPHONY [Quinn 2001] uses multiple instruments to sonify aspects of climate, such as ice sheet movements and volcanic activity—each of these are mapped to various instruments and together these instruments produce a symphony, which premiered at the American Museum of Natural History. Recently, CERN's ATLAS AUDIO [Hill and Goldfarb 2016] and LISTENING TO THE MIND LISTENING [Barrass et al. 2006a; Barrass et al. 2006b] have served a similar idea of presenting scientific data in audio format to make it more accessible for the general public. LISTENING POST [Hansen and Rubin 2001; 2002] is a piece of art that sonifies internet discussions using a large amount of data collected from several online forums. It uses text-to-speech technology to play selected words and sentences from these discussions, and has been exhibited, e.g. in the Brooklyn Academy of Music in 2001.

Previous work on sonification has also touched topics like emotions and short textual messages, relevant to our demonstrators in the coming sections. A recent example is sonification of footsteps to give an impression of the particular mood of the walker [Turchet and Rodà 2017]. They found, e.g., that aggressive emotions—expressed via faster walking speed—can be differentiated from happy emotions. The sonification can also be interactive and can have an impact on the walking patterns [Turchet and Bresin 2015].

SkypeMelody [Alt et al. 2010] is an interesting example of melodic sonification used to relay information. It transforms instant messages or SMS text messages into an aural format to allow easy apprehension of basic message content. SkypeMelody pays special attention to emoticons, selected keywords and punctuation, and transforms the data to a simple melody that represents the content of the message. The methodology not only maps words to notes but also, for example, chooses the scale of the melody based on sentiment. The amount of musical knowledge is low and the mapping from data to sound is quite straightforward.

3. THE DATA MUSICALIZATION PROCESS

We next outline the process of data musicalization. Since the processes can vary greatly across applications, we do not aim to give a definitive description but rather highlight issues that need to be addressed and give examples based on our own work with the applications of Sections 4–6.

We identify three aspects to building a data musicalization application: *musical design* focuses on the sofware designers' decisions about the aim of the data musicalization and the genre of the music ('aim' in Table II), *automated composition* is about the software actually producing the music from data ('composition methods', related also to 'amount of musical knowledge', 'way of obtaining musical knowledge' in Table II), while *external engineering* is concerned with other system aspects, such as the input to the musicalization methods and evaluation of the musicalization system ('input data' in Table II).

3.1. Musical Design

The musical design process involves decisions regarding the aim of the musicalization and is critical because the goal of musicalization is not purely to convey information. This process answers the questions of what type of musical experience should be created and what type of limitations are part of that creation. The aim can vary from ambient transfer of information to artistic expression. Similarly, musical considerations can affect the operating mode of the musicalization—batch, real-time or even interactive—as this can reflect different types of artistic expression of the data. Similarly, the choice of using unimodal (audio only) and multimodal (audio and other complementary mediums) is again related to the artistic aim of the musicalization and will be reflected in this process.

While the focus on musical or even artistic aspects may feel unnatural for computer science in general, this is certainly not the case for the subfield of computational creativity [Cardoso et al. 2009; Colton and Wiggins 2012]. However, it is difficult to offer any general guidelines for making these artistic decisions. Colton et al. [2011] argue that in making such decisions, one needs to know the intended audience; thus, the musical design for musicalization should be aligned with the expectations of those who will listen to the results of that process.

3.2. Automated Composition

Mechanistic composition of music is an old idea, dating back almost one thousand years to Guido d'Arezzo, if not earlier [Kirchmeyer 1968]. Well-known contributions in the area also include Mozart's Musikalisches Würfelspiel. The first known computer composition, The Illiac Suite for String Quartet, was generated in 1956 by a program of Hiller and Isaacson. Current state-of-the-art methods include David Cope's program called Emily Howell which composes complex music in its own style [Muscutt 2007].

Modern computer-based methods use a wide variety of techniques to compose music, including fractals, grammars, constraints, pattern matching, and various stochastic processes such as Markov chains. For an overview of the basic techniques, see Roads [1996] and Fernández and Vico [2013]. The challenges of music generation are also known to the multimedia community. Lin et al. [2015] constructs musical medleys from users' music libraries. While the task in this case is not to *compose* music, it is related in the sense that the challenge is still automatic creation of a new musical artifact.

Because there are likely as many approaches to composition as there are composers, there is no accepted "correct" way to approach the problem. To illustrate one way of structuring the task of musical composition, we next describe how we approached the problem in several of the applications of the following sections. Many other approaches are equally viable (see references above), and even our applications following this design vary the order of the stages. Also note that the techniques discussed here all focus on only local structure. Coherence and high-level structure are still difficult problems in automatic composition, and we do not offer any solutions for them in the work described here, other than occasionally specifying some aspect of them *a priori*.

Generation of Harmony. One simple approach to harmonization, i.e., production of chord sequences, is to use a manually crafted first-order Markov chain. The transition probabilities for chords can be chosen to reflect their frequencies in, for example, Western music; the transition probabilities could also be learned from given music. To reduce the complexity of the model, it may be restricted to a small set of possible chords (such as the seven elementary ones). A slightly more complex model might incorporate higher order statistics (an order 2 or greater Markov chain, for example). Depending on the desired model complexity and availability of data, not all higher-order states might have stable, non-zero transition probabilities, and in such cases, it is possible to use a lower-order Markov chain as a backoff model.

Another approach to implementing this stage is to use a Hidden Markov Model (HMM) to model sequences. In one of our implementations, hidden states represent 24 different chords (12 major and 12 minor), and the observation alphabet consists of 12 pitch classes. Such an HMM requires two conditional probability distributions: the probability of a melody note given a chord and the probability of a chord given the previous chord. The statistics for these probability distributions can be estimated by analyzing chords and melodies from a corpus of musical examples. Given this HMM, we can harmonize on a generated melody using the Viterbi algorithm to find the chord sequence most likely to accompany the extracted melody (note that in this case generating the chord sequence requires that the melody has already been generated).

Generation of Melody. A simple yet effective approach to melody generation can be effected by relying on a previously generated underlying chord sequence. Given such a chord sequence, a melody can be composed by considering two things: (1) the internal moves within the melody line (intervals between successive notes), and (2) the relationship of the melody to the current chord.

Intervals can be generated again using a Markov chain. To control the magnitudes of movements in the melody, the model can be manually constructed to favor small intervals, which is usual for Western music, and to prohibit melodies that ascend or descend too rapidly.

To produce a melody that matches the underlying chord sequence, an initial note is (randomly) sampled from the underlying chord. Then, the interval model is sampled and the obtained interval is added to the previous note. If the resulting note is not in the appropriate scale or if it is dissonant with the current chord, it is discarded. In such a scenario, the Markov chain is resampled to generate a new interval instead.

Another approach to melody composition, which does not depend on an existing harmonization, is simply to draw melodic pitches from a Markov chain (i.e., an n-gram model) constructed from the melody lines of music in an appropriate corpus. A melody is initialized with a series of random notes, selected from a distribution that models which notes are most likely to begin musical selections in the corpus. Additional notes in the melodic sequence are randomly selected based on a probability distribution of what note is most likely to follow the given series of n notes. Such a system can be used to generate several hundred candidates for a rhythmic phrase, and each candidate can then be evaluated in some way (e.g., by a neural network trained to discriminate musical affect or to model user preference or musicality) and the best one incorporated into the composition.

A third alternative might use the HMM discussed above to simultaneously generate both harmony and melody—the hidden state is sampled to produce the current chord, and then the observed state is sampled to produce the current note.

Generation of Rhythm. A simple approach to the generation of rhythmic sequences is the use of pre-defined templates. Another template-based approach is to sample rhythmic sequences from a musical corpus. A slightly more complex approach would be to model the sequences, again using a Markov chain. Transition probabilities can be set manually to favor certain rhythmic patterns (such as repetition of short durations) or learned from a corpus. Further, transition probabilities might be dynamic, so that rhythmic patterns are more likely to match a time signature (for instance, if the last note is 1.5 beats long, the next one is likely to be 0.5 beats long). Additional constraints can be used to enforce a reasonable sequence ending pattern.

Generation of Accompaniment. In the simplest case, accompaniment decisions (harmonization patterns, voicing, instrumentation choice, etc.) can be made *a priori*. However, it is often more interesting/useful to make them data-dependent. Accompaniment choices can radically alter the tone and feel of a musical piece, and if these can somehow reflect characteristics of the data or aestethics (or both), musicalization is typically more successful than if they can not. In such situations, this stage of composition is typically intimately tied to the artistic process discussed above.

Generation of Dynamics and Tempo. As with accompaniment, both dynamics and tempo have a significant impact on the artist effect of a musical piece (in addition, both can be used in communicating information as well). Both can be chosen *a priori* and held constant or can be a function of some aspect of the data or incorporated into a model (e.g. some variation of a Markov model, recurrent or non-recurrent neural network, etc.).

3.3. External Engineering

We address two non-musical technical aspects of musicalization applications: input data and evaluation of the musicalization system.

Input data and data extraction. While the process of data musicalization is often neutral to the data source (i.e., any data can be used in this process), in this work we have focused on two broad categories of data source that have garnered significant interest: sensors and text [Grimmer and Stewart 2013].

Musicalization of sensor data has the potential to allow perception of multidimensional data streams by encoding those data and/or their properties in novel formats, giving them new meaning for the user. Sensory data, however, requires preprocessing, e.g. noise removal, normalization and other forms of cleaning.

Similarly the preprocessing of textual data includes operationalization of the text into meaningful variables: this can, for example, be based on bag-of-words approaches such as LIWC [Pennebaker et al. 2007] or SentiStrength [Thelwall et al. 2010] or more advanced text mining and machine learning tools, like topic models. Here again, the 'raw' data is preprocessed into a more meaningful format during this process.

As seen in these examples, the key aspect is not the data, but rather the ability to extract meaningful features and information from the data. One interesting way of translating features of data to music is via emotions. Juslin [2001] has mapped emotions such as happiness, fear and sadness to musical attributes such as tempo, tone attacks, and timbre. Similarly, Bresin and Friberg [2011] observed a relationship between tempo and activity, and between articulations and emotional presentation. And, Monteith et al. [2010a; 2010b] have built a music composition system to communicate a given emotion. While results such as these allow the composition of music to express emotions, the question of how to map data to emotions remains application-specific.

Evaluation of a musicalization system. The chosen evaluation method should relate to the developer's aim for the system. In cases where the aim is to convey information, objective measurements can be used to conduct this evaluation. For example, Bonebright and Flowers [2011] suggest several methods that can be used to evaluate auditory displays and audio interaction, including accuracy (meaningfulness), experience (pleasantness) and discrimination. Previous studies on computer-generated audio have also focused on how much subjects liked the music [Juslin et al. 2002]. Where applicable, our work evaluates the success of the mapping primarily by measuring the accuracy with which participants correctly identify the mapping between data and music without being informed of the mappings used, but we also measure how pleasant they think the music is. As mentioned above, this manner of evaluation is applicable mostly when the aim is transfer of information.

If the aim of the data musicalization system is artistic expression or invocation of subjective emotions, measures based on accuracy are irrelevant. Instead, the creative output should be evaluated against the creator's intentions and artistic goals. Some authors in the field of computational creativity go even further and argue that it is not so much the creative artefact (music) that should be evaluated but rather the creative process itself. FACE [Colton et al. 2011] is a prominent model for this, considering four different types of creative processes and products:

— Generation of new *framings*. Framing is a piece of information which explains or justifies the whole creative act. In musicalization, the input data provides a partial explanation of the result. No framing is generated, however, and the role of data is better described as affecting the aesthetic measure, as described below.
— Generation of new *aesthetic measures* (cf. Table I). An aesthetic measure indicates what the system considers good output and what it aims to produce. In musicalization, this is typically fixed by the system designer or (partially) learned from example music.
— Generation of new *expressions* (cf. Table I). In musicalization, the expressions are music and they are by definition always generated by the musicalization system.
— Generation of new *concepts*. Concepts are more abstract types of expressions. Like aesthetics, in musicalization, these are typically fixed by the system designer.

4. SLEEP MUSICALIZATION

In this and the following sections we describe musicalization applications.

In this section, we illustrate possibilities of data musicalization with a number of different approaches and techniques to sleep data musicalization (cf. Table II). Modern sensors allow fully unobtrusive sleep measurement in one's own bed, potentially helping anyone improve their sleep and well-being by measuring and tracking sleep over extended periods of time. Sleep musicalization, an application we have previously introduced [Tulilaulu et al. 2012], aims to add a fun factor to sleep tracking by providing affective perceptualizations that complement more traditional representations of sleep measurements. In addition to this original application, we describe three novel ones that use sleep measurements as inspiration and aim more at artistic results. Sample music files produced with the methods from real sleep data are available in an Online Appendix.

We start the section with a very brief overview of how we perform sleep analysis, in order to make this paper more self-contained. We have described the sleep analysis methods in detail elsewhere [Paalasmaa et al. 2011; Paalasmaa et al. 2012; Paalasmaa et al. 2015].

4.1. Sleep Analysis

Sleep is usually divided into five sleep stages: wakefulness, REM (rapid eye movement) sleep, and three categories of non-REM sleep: N1, N2 and N3 [Iber et al. 2007]. Stages N1 and N2 are called light sleep and N3 deep sleep. Medically, sleep stages are defined using *polysomnography* which involves biopotential electrodes attached to the head, and which costs hundreds of euros per night.

A common sleep measurement method is actigraphy, where the user wears a wrist accelerometer. In medical use, the duration of a study typically is one week, but with the rapid growth of wrist activity sensors in the consumer market, possibilities of and interest in sleep analysis have been growing. Unfortunately, actigraphy has limited diagnostic capability and cannot be used to detect sleep stages.

We use a modern, commercially available Beddit mattress sensor[2] that can be used to detect sleep stages as well as respiration, heart rate and movements.

Sleep Data Collection. The Beddit sensor obtains its raw signal using a thin, flexible, piezoelectric force sensor. The sensor measures 70 by 4 cm and is placed under the mattress topper or mattress. The sensor is highly sensitive to changes in the force directed at it; signal from the sensor is sampled at 140 Hz.

The force signal detected by the sensor, for a person lying still, essentially consists of forces caused by the person's heartbeats and respiration, as well as noise. Using signal analysis methods described elsewhere [Paalasmaa et al. 2011; Paalasmaa et al. 2012; Paalasmaa et al. 2015], we detect the heart rate, the respiration rate, and variability in both of them. Sleeper movements are detected by analyzing individual abrupt changes in the force signal.

[2]http://www.beddit.com

Our sleep stage analysis is largely based on heart rate and respiration variability as well as movement information [Paalasmaa et al. 2012]. For each night, the time spent in bed is segmented into periods of wakefulness, REM sleep, light sleep and deep sleep, i.e., the *hypnogram*. We also record the heart rate, respiration rate, and movements of the sleeper during the night.

4.2. Musical Perceptualization of Sleep Patterns

In Sleep Musicalization [Tulilaulu et al. 2012], a novel piece of music is automatically composed from the sleep measurements of a single night in order to perceptualize the hypnogram and some other characteristics measured from the sleep. The goal of sleep musicalization here is to produce a piece of western classic style piano music to reflect the sleep measurements.

As input, we use data on sleep stages, heart rate, and movements, available from the beddit.com service. Before the actual music generation, we carry out some preparatory actions. First, short sleep stages are removed to give more stability to the resulting music. Second, a major musical scale is chosen at random. Third, the length of the bars is decided randomly between 3 or 4 beats, in order to produce more variance in the style of the music. The measurements are then mapped to properties of the generated music as follows.

— Each sleep stage has a unique accompaniment pattern reflecting the depth and peacefulness of sleep. This gives each sleep stage a distinctive style and feeling in the music.
— Tempo (beats per minute) reflects the heart rate. However, the heart rate is transformed to make changes more pronounced.
— The rhythm (average lengths of notes) varies slightly with respiration rate.
— The volume of the music reflects the density of sleeper movements.

Details of the method can be found elsewhere [Tulilaulu et al. 2012].

We produce a highly compressed representation of the measurements, where one second of music corresponds to two minutes of sleep, i.e., eight hours of sleep results in a song of about 4 minutes. The sleep musicalization method is available as a web application[3]. With it, sleep music can be generated and accessed conveniently by users of the Beddit device. The Python sources for the musicalization[4] and web service[5] components are available for download. We use Kunquat[6] for the actual sound generation, given the composed music.

The goal of sleep musicalization is to provide a novel way to perceptualize sleep measurements, complementing their visualizations. According to our subjective evaluation, the music produced by the sleep musicalization method above reflects the stages of the sleep cycle quite naturally and the sleep stages' different natures are clearly audible in the music. When compared to the visualization of the same sleep measurements, the music contains less details as the shortest sleep stages have been stripped out. Otherwise, the same elements can be clearly found in both representations. Sleep is mostly represented as quite peaceful music, and deviations from this are justified by data: lots of movement or REM sleep stage.

The music is in general quite listenable. It resembles background music: it does not have many characteristics that draw attention, and it is not irritating, except possibly for periods of bad sleep. The standard Western chord progressions sound natural to us. The melodies are a little odd, as they don't have the repetitive elements that music often has, but they do make sense in the context of the chords that they are played with.

Formal user studies of sleep musicalization are difficult: sleepers themselves could mostly try to correlate the music with the sleep analysis since sleepers are otherwise unaware of their sleep stages. Instead, we invite readers to form their own opinions by listening to sleep music using the Online Appendix or at http://sleepmusicalization.net. Independent reviews of sleep music are partially favorable: for instance, Leslie Katz of CNET[7] writes that "*they do have a uniquely personal and (dare I say it?) dreamlike feel*", while Nic Halverson of Discovery News[8] wrote that "*these songs left me with a sense of incompletion, as if the piano-dominate* [sic] *music would be an inadequate representation of my dreams.*"

[3]http://sleepmusicalization.net
[4]https://github.com/Tulilaulu/Sleep-musicalization
[5]https://github.com/beddit/sleep-musicalization-web
[6]http://kunquat.org
[7]http://news.cnet.com/8301-17938_105-57511480-1/your-sleep-patterns-now-in-soundtrack-form/, Sep 12, 2012.
[8]http://news.discovery.com/tech/music-made-from-dreams-120912.htm, Sep 12, 2012.

4.3. Automatic Music Composition from Sleep Measurements

In an effort to combine sleep data perceptualization with musical aesthetics, we implement three modular music composition systems that interact with the sleep musicalization system described in Section 4.2. The first system uses a raw force signal from the Beddit sensor. The signal is compressed in time to give an audio signal that is used for melodic "inspiration." The second system composes melodies based on emotional cues for each sleep stage and harmonizes the melodies using hidden Markov models. The third system creates musical motifs from the raw force signal, combines the motifs into a melody using a genetic algorithm, harmonizes the melody using a hidden Markov model, creates a music lead sheet, and performs the lead sheet. Sample compositions can be listened to using the Online Appendix or at http://mind.cs.byu.edu/music/sleepMusic/.

4.3.1. Melodic and Harmonic Music Generation Inspired by Sleep Measurements. The first system composes music that is "inspired" by the sleep data similar to the approach used by Smith et al. [2012]. The force sensor signal is requested from the beddit.com service, and the composition process begins by extracting pitches for the melody from the signal data. The sample rate of a sleep signal is converted from 140 Hz to 44.1 kHz, making the sleep signal shorter and more audible, and the DC offset (i.e., mean of the waveform) is shifted to zero. The primary components of the force signal are respiration (∼0.2 Hz), heart rate (∼1 Hz) and the heartbeat complex (∼10 Hz). Because the conversion multiplies the sample rate by 315, those phenomena become audible. Pitch detection is performed on the resulting audio signal with a command line utility called Aubio[9], and the detected pitches form the melody in the final composition. Optionally, tonality is enforced by randomly shifting non-diatonic pitches up or down by a half step, where the scale with the fewest necessary pitch shifts is chosen.

We use a Hidden Markov Model with 24 states, trained with chords and melodies from 402 Bach chorales, to harmonize on the generated melody. The rhythm and dynamics for melody pitches and chords are obtained using the sleep musicalization methods described in Section 4.2. The resulting melody and accompaniment are then combined into a composition using jMusic[10], an open source music composition library. The final composition is written to an audio file using a virtual piano.

According to our subjective preference, the inspired compositions with enforced tonality are quite listenable. The sequences of melodic pitches are mostly conjunct with occasional disjunct motions. A variety of rhythms appear in the melodies, and this results in a nice blend between syncopation and straight rhythm. The accompaniment chords generally fit the melody closely, but the chords are a bit repetitive and predictable. Common chord progressions naturally emerge, and cadences often involve strong relationships between tonic, dominant, and subdominant chords. Accompaniment patterns are simple and they clearly outline each chord choice. Overall, these compositions resemble music that might be found in a method book for beginning or intermediate piano students.

The compositions without enforced tonality are not as easy for the average listener to absorb, but the content is perhaps more intellectually interesting. Except for a few distinctions, these compositions are fundamentally similar to the ones with enforced tonality. The music is more dissonant, which typically corresponds to greater discomfort for the listener. The simple accompaniment patterns and chords do not mesh as well with the dissonant melody. However, certain composers and listeners might be intrigued by this effect. This method could be improved by merging the interesting portions of the atonal compositions with the pleasing portions of the tonal compositions. We leave it to the reader to listen to the sample output in order to compare and analyze the qualities of these compositional methods (see Online Appendix).

4.3.2. Emotionally Targeted Music from Sleep Measurements. The second artistic system creates music to communicate an emotion, using a system built by Monteith et al. [2010a; 2010b] designed to compose music that targets one of six basic emotions: anger, joy, fear, love, sadness and surprise [Parrott 2001]. That system is adapted here to compose music in segments that correspond to sleep stages. Each sleep stage is converted to an emotion using a simple mapping, with an emotion being chosen at random in the case that the mapping yields multiple values (see Table III).

For each sleep stage, we then generate a melody, add harmonizing accompaniment and choose instrumentation as follows. For each of the six emotions, we employ a corpus of music (movie soundtracks) specifically labeled by human volunteers as being expressive of the targeted emotion. For each corpus, we train a neural network model to discriminate music targeting that emotion from music targeting any of the other five emo-

[9] aubio.org
[10] explodingart.com/jmusic/

Table III. Mapping sleep stages to basic emotions.

Sleep Stage	Emotion
Awake	Love, Joy
Light	Fear
REM	Anger, Surprise
Deep	Sadness

The four measured sleep stages are mapped to six basic emotions, which provide input to an HMM-based automatic composition system. The emotional target for generated music changes with measured sleep stage changes.

tions (six networks total). We also train a seventh network to discriminate music created by the system from music in any of the corpora (i.e. to identify music already accepted by human audiences). These networks are used to evaluate the efficacy of generated melodies, giving the system some level of appreciation for what it creates. Rhythms and dynamics are again composed by the sleep musicalization methods described in Section 4.2.

After composition of a complete segment for each sleep stage, all the segments are concatenated to form the entire piece, which communicates a series of emotions based on the experienced sleep stages, and the final composition is written to an audio file, using jMusic. Note that though emotional communication through music is a very subjective experience, Monteith et al. [2010a; 2010b] have shown that music created in this way is reasonably successful at communicating emotions to human listeners and is comparable, in this respect, to music generated for the same purpose by (amateur) human composers.

Despite the difference in compositional algorithms, the melodies produced by the emotional system are similar to those produced by the inspirational system. This is partially because both systems use the same rhythm and dynamics generator of Section 4.2. The melodic pitches chosen by the emotional system sound slightly more musical than those from the inspirational system. This is certainly the case because the composition of the pitches is based on learning from a corpus, whereas the inspirational system composes melodies from a raw sleep signal without any training. The emotional system also produces more rests in the melody than does the inspirational system.

In addition, the accompaniment patterns produced by the emotional system are richer than those from the inspirational system. This gives more contrast between each sleep stage, and the transitions between stages are more recognizable. Although it is nice to aurally distinguish the sleep stages from one another, the transitions between stages are often jarring. In future work, these transitions could be smoothed using a number of techniques.

4.3.3. Motivic Composition using Sleep Measurements. The third artistic system creates a lead sheet by extracting a sequence of notes from a sleep signal using a novel technique, discovering musical motifs from the sequence of notes, creating a melody from the discovered motifs using a genetic algorithm, and harmonizing the melody.

In order to extract a sequence of notes from a sleep signal, we pre-process the signal by shifting the DC offset to zero. An example of a pre-processed sleep signal is shown graphically in Figure 1. Next, we isolate each section of samples that are above zero and extract a note from each of these sections. The note duration (in 32nd notes) is determined by taking the square root of the number of samples in the section and converting it to an integer by using the floor function. The note pitch is determined by calculating the maximum amplitude in the section modulo 30 and adding it to midi note number 55. Thus pitches are in the midi note number range [55,84]. This range is somewhat arbitrarily chosen as a suitable range for musical motifs. Figure 2 shows the peaks from a signal that would be used to determine the pitches in a note sequence.

After the string of notes is detected, we discover musical motifs according to a method described by Johnson and Ventura [2014]. We create an initial population of 100 themes by repeatedly choosing 4 random motifs and concatenating them. We run a genetic algorithm with the initial population for 100 iterations without a stopping criterion, assuming a 4/4 meter. The fitness function, crossover, and mutation operations are taken from the NormalDistributionFE, OnePointCrossover, and ComplexMutater classes in jMusic, respectively.

The genetic algorithm above is executed twice in order to create an A theme and a B theme. The A and B themes are then concatenated to create a melody line in ABAB form, a typical form in pop music and jazz.

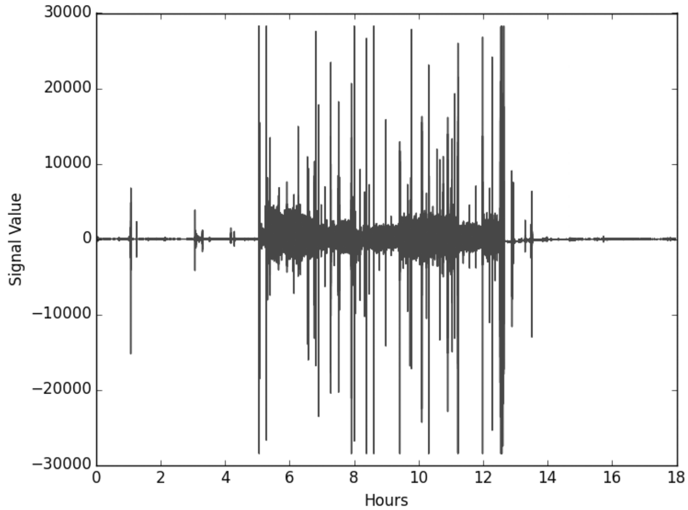

Fig. 1. An example of a raw sleep signal after shifting the DC offset to zero. The bursty period in the middle of the graph corresponds to some 9 hours of sleep.

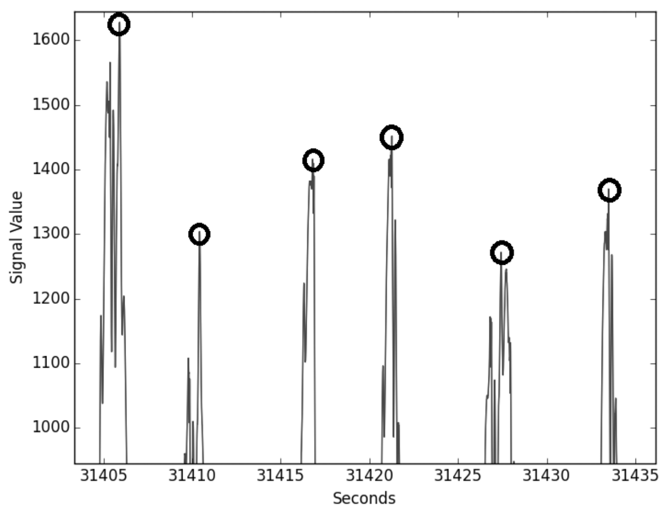

Fig. 2. Peaks from a sleep signal. This is a zoomed-in version of Figure 1. Pitches are determined by selecting the peaks from each positive region of the signal, and durations are determined by the number of samples in each positive region. The peaks in the signal are circled; they occur at approximately 6 second intervals, corresponding to respiration cycles.

After the melody line for the entire piece of music is created, we add chords to the melody by following the process described by Simon et al. [2008]. We collect chord transition probabilities in a square matrix by counting chord transitions in a chords-only lead sheet database called *The Imaginary Book*, which can be downloaded at the Impro-Visor Yahoo Group.[11] We generate melodies for *The Imaginary Book* with Impro-Visor's improvisation tool using the Bill Evans grammar. Then we collect melody emission probabilities (the probability that a given chord accompanies a given pitch class) in a matrix by counting the melody note durations for each accompaniment chord.

[11] https://groups.yahoo.com/neo/groups/impro-visor/info

We segment our composed theme into measures, where each measure becomes an observation in a hidden Markov model, and the chord transition matrix represents the state transitions. The measure observations are calculated by taking the dot product of the appropriate row in the melody emission matrix with the pitch vector that represents the measure, and this constitutes the emission matrix of a hidden Markov model. The Viterbi algorithm is used to calculate the most likely chord sequence that accompanies the given melody. Simon et al. [2008] provides precise implementation details.

When compared to the systems of Sections 4.3.1-4.3.2, the output of this system is perhaps the most similar to music composed by humans. This is largely due to many elements in the system being hand-picked by its creators rather than by the system itself (e.g., form, range, and meter). The fitness function, crossover, and mutation operations from jMusic are also restrictive, forming melodies that are mostly tonal and managed. While the genetic algorithm outputs pleasant melodies, it is difficult to discern any inspiration from the motifs that were given as inputs.

The chords chosen by this system are less exploratory than the chord sequences in the HMM training set. For example, there are more I, IV, and V chords and fewer ii, iii, vi, and vii chords produced by this system than by one based on the training set. One explanation for this outcome is that the melodies on which the training set is based are more exploratory than those producible within the constraints of this system. If this system allowed, for example, melodies that contained more chromatic movement, then perhaps the system would output a richer variety of chords to accompany the melodies.

5. CHAT MUSICALIZATION

We next describe an application of data musicalization to online chat sessions. We extend the text-based interaction of a chat discussion by musicalization of some of its key characteristics, aiming to enhance the perception of the rate of messages, of the number of active participants, and of the tone of voice (valence). The goal is that participants could follow key characteristics of the chat without reading the content in detail. For example, a lecturer may listen to ambient information from a background discussion between students; or users could follow general activity in social media while primarily occupied by other tasks. Therefore, chat musicalization here focuses on transmitting information, akin to the first sleep musicalization example in Section 4.2. With this goal in mind, we conducted an empirical evaluation on users' ability to follow the music and link it to the chat discussion.

Mapping Chat to Musical Parameters. We use music generation methods similar to those described in Section 4.2 for sleep musicalization. For brevity, we focus on the mapping between the chat and the musicalization parameters. We use mappings that have been found successful in previous research, as follows.

Tempo is a natural indicator for activity [Juslin 2001; Bresin and Friberg 2011]. Therefore we used the lag between messages to adjust the tempo: when messages come with longer delays between them, the tempo of the music is also lower, varying between 20 and 117 bpm.

An important characteristic of a chat is the number of active participants. We mapped the total number of participants to the number of voices used in the music generation. 1, 2 and 3 participants are played by 3, 4, and 5 voices respectively. Use of just 1 or 2 voices was avoided as it produces musically less satisfying results.

The musicalization also encodes certain aspects of the content of the messages through the use of harmony and scale: messages with positive sentiment produce consonant music played in a major scale; whereas messages with negative sentiment lead to cacophonic music played in a natural minor scale. We used SentiStrength [Thelwall et al. 2010] to conduct the sentiment analysis. To enhance continuity of the music, we used a sliding average over the three latest messages' SentiStrength values.

In this application, musicalization takes place in real time. The system thus cannot plan ahead much and has to be able to adapt swiftly to changes in the input. Our subjective view is that the input variables are hearable in the music the program produced (for empirical evaluation, see below). The most notable parameters are for us easy to spot: the number of instruments, the tempo and the dissonance of the music. Examples of this music are provided in the Online Appendix.

The music sounds, by design, like background music without notable characteristics. The music is pleasing, except when made dissonant due to a negative sentiment. It is hard to follow, however, because of the lack of structure. The instruments all move separately on an agreed chord, but without a clear destination. This is largely due to the real-time constraint on the system that makes planning difficult.

Evaluation. We compare the musicalization method proposed here to an alternative method by Alt et al. [2010]. It sonifies linguistic characteristics of short text messages (SMSs), allowing users to perceive them

Table IV. Means and variances for three types of musicalizations, and statistical significances of the differences to the musicalization proposed in this paper

Question	Musicalization (this paper) ave (var)	Musicalization [Alt et al. 2010] ave (var)	signif	Musicalization of a different text ave (var)	signif
Q1. I think the music was pleasant	3.32 (1.56)	3.71 (1.21)	n.s.	2.31 (0.90)	$p < 0.05$
Q2. The music was interesting	3.86 (0.98)	3.43 (1.16)	n.s.	2.29 (1.26)	$p < 0.05$
Q3. The music sounded natural	2.95 (1.57)	3.05 (1.15)	n.s.	1.94 (0.86)	$p < 0.01$
Q4. Hearing the music made it harder for me to follow the chat	2.68 (2.42)	2.67 (1.93)	n.s.	3.06 (1.80)	n.s.
Q5. The music reflected the chat discussion	3.09 (1.13)	3.38 (1.55)	n.s.	2.44 (1.60)	$p < 0.1$
Q6. The music matched the content of the conversation	3.14 (1.46)	2.86 (1.13)	n.s.	2.26 (1.85)	n.s.

(n.s. = not significant).

without interruption. It is thus applicable also to sonification of chat messages, and has a similar aim of non-interruptive observation of textual content.

The evaluation was conducted in laboratory settings with upper secondary school students ($n = 24$), age 17 to 18. The participants had mixed previous experience of (playing) music: 9 had no musical experience, 9 had played an instrument on their own and 6 had played in a band or in a performance. During the study, participants followed a chat discussion and at the same time heard music. After each of the chats, we asked them to evaluate the quality of the music and the relationship between the music and the text.

We used three different samples of text from existing research on chats to ensure they were natural[12]. Each participant observed each of the chats with some background music, an accompaniment of one of three types: (1) music from our system, (2) music generated by the alternative system [Alt et al. 2010], and (3) music generated by our system for a different chat. The order in which the generative systems were used and the order in which chats were presented was randomized per participant to control for learning effect and fatiguing. Each of the examples lasted equally long, 4.5 minutes, but they varied in their number of comments and tempos. The total duration of the experiment was therefore about 20 minutes to limit fatigue.

After experiencing a chat discussion enhanced with music, the participants evaluated, using a five-point Likert scale (Strongly disagree - Strongly agree), how pleasant the music was and how well the music reflected the chat discussion (see left column of Table IV for the exact questions). The questionnaire was presented to the participants before the first experiment to ensure that the questions were understood. Otherwise, the subjects were not informed about our research questions or about the mappings used (e.g., that a larger number of voices would correspond to a larger number of active chat participants). Any perceived links between musical features and chat characteristics are thus independently recovered by the subjects. While this is an interesting test of the power of data musicalization to describe the data, it also makes the setting challenging for users.

The results of the evaluation are presented in Table IV. We report the means and variances for each of the six questions, for each of three different musicalizations. We also report the statistical significances of differences between the musicalizations (ours vs. the alternative one, and ours vs. musicalization of a different text) using the two-tailed Wilcoxon, or Mann-Whitney, test. We also confirmed that there is no ordering effect by examining the grades each musicalization received when it was heard first, second or third. Similarly, the musical skill-level of the participants did not affect the evaluation.

The two different musicalizations produced from the chat text produce similar results. The music from our system is considered slightly more interesting (Question Q2) and a better match to the conversation (Q6), while music from the alternative system is slightly more pleasant (Q1) and natural (Q3), and better reflected the discussion (Q5). All these differences are small, however, and not statistically significant. The means for both of these musicalizations are mostly around three, which indicates neither agreement or disagreement with the questions. Obviously, both methods have room for improvement in representing chat discussions as music.

[12] Taken from [Curtis 2004] and [Bradner et al. 1999]

However, a more important result is that the ratings for the two musicalizations of a chat text are significantly better than the ratings for musicalization of a different text (Q1–Q3 and Q5). This shows that there are observable differences in how well a musicalization reflects the discussion (especially Q5). For questions Q4 and Q6 that also relate the music to the text, similar differences exist but they are not statistically significant.

An interesting, unexpected result indicates that the observed quality of music depends on how well it matches the text: questions Q1–Q3 show that the music generated from a different text was considered less pleasant (Q1), less interesting (Q2), and less natural (Q3) than music generated from the current text. Interestingly, however, questions Q4 and Q6 about the match between music and text did not reflect statistically significant differences. These results suggest that the perceived quality could subconsciously depend on the match between data and music, even when the perceived differences in that match are not significant.

6. INTERACTIVE MUSICALIZATION OF GESTURE DATA

As our final example of data musicalization, we introduce a completely different, interactive case. *Musicreatures*[13] utilizes the multimodal interaction capabilities of smart phones in playful, real-time music creation. The goal of the application is that the user, regardless of their musical background, can create music in a simple game-like environment mainly using motion gestures.

Using mobile devices as musical instruments has been explored both before and, crucially, after the advent of modern smart phones, such as the iPhone. The viewpoints include, for instance, an analysis of the sensor capabilities [Essl and Rohs 2009], a repertoire-based mobile phone ensemble [Wang et al. 2008; Oh et al. 2010], and musical, social interaction geared towards the masses [Wang et al. 2009]. With Musicreatures, we have placed the emphasis on a rather special combination of approaches: firstly, designing an application the use of which requires virtually no learning or musical expertise, secondly, providing the user creative authority over the resulting musical composition, and lastly, refraining from substantial use of the touch-based interaction method. This paper is the first publication about Musicreatures.

We have designed Musicreatures to highlight several interesting points for data musicalization. First, parallel to the previous applications, while more direct sonification of gestures could be used to produce sound, the concept of data musicalization as composition of music "inspired" by the gestures can guarantee more pleasant results. Second, however, the application works interactively in real time and should be responsive to user actions.

We have aimed at a useful combination of responsiveness and musicality by adding a layer of visual representation between the gestures and music. Gestures control a visual view containing simple objects, and the visual data is then musicalized. This way, the user gets immediate feedback from the visual view, while the visual view and data musicalization derived from it enforce a number of musical restrictions in order to produce musically satisfying results.

In general, translating gestures into music that sounds good is difficult. An obvious direct approach would involve turning kinetic data into quantized rhythmic and melodic features that sound good together. The major issue is that temporal quantization induces lag between the triggering gestures and the resulting events, preventing strictly real-time musicalization. Levitin et al. [2000] found that a delay of just 40 milliseconds in producing a sound that relates to an event observed via touch or sight causes a perception of cross-modal asynchrony. Assuming a relatively fine-grained grid based on a tempo of 110 beats per minute and a 16th note quantization region, a naive real-time quantization using the ceiling function causes delays of over 130 milliseconds. Additionally, the delay depends on the timing of the triggering gesture: this makes it difficult for the user to compensate for the lag between a gesture and its feedback.

Musicreatures has two modes that solve the gesture musicalization challenge in different ways: the standard mode for visually mediated musicalization of gestures and the improvisatory mode for more direct musicalization of gestures.

Standard Mode. In the *standard mode*, the user controls a simple visual scene of circular objects by tilting and moving the phone, i.e., kinetic information from the user is transformed into visual movements of the objects (Figure 3). At the same time, the positions and states of the visual objects are used as the input to musicalization, transforming the information from the visual domain to the auditory domain.

Musicalizing the states of the objects represented visually on the screen enables the auditory feedback to have temporal flexibility, circumventing the quantization problem. There are two aspects to this. First,

[13]Available for the Apple iPhone at App Store free of charge (https://itunes.apple.com/us/app/musicreatures/id957316801).

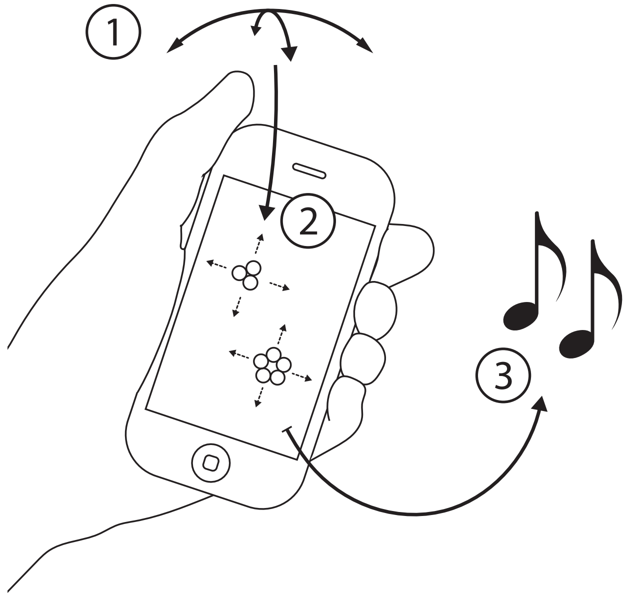

Fig. 3. Interaction modalities and flow of control in the Musicreatures application. 1. Kinetic input by tilting the phone; 2. Movements of visual objects as intermediate feedback; 3. Music created from the visual data.

the visualization gives immediate feedback to the user's actions, showing that the user is in control of the application. Second, the visualization illustrates that the timings of notes are regulated, showing how they are independent of the immediate kinetic input.

Musicalizing the visual scene instead of direct user gestures thus affords controlled musical results while the user is still given substantial creative authority. The musicalization of the visual scene has been designed to result in consistent rhythmic accuracy; the user is only given coarse, indirect control over the timing of the musical events. Similarly, the result is harmonically consonant since the user does not have direct control over the pitch attributes. We next describe the musicalization process in some more detail; it is tailored for this mobile application and differs from the methods used in the previous sections.

When the user tilts the device, the on-screen objects move correspondingly. These visual objects—considered individual instruments as regards the musicalization—change their colors according to the colors of the background image. The objects' saturation and brightness are mapped to pitch parameters, but a change in pitch is not audible before the object in question plays its next note (the objects play non-continuous, percussive notes). Again, the visual feedback has a vital part in confirming user actions: it indicates immediately that the state of an object has changed.

Rhythms are treated as follows in the musicalization. Each user-controllable visual object—a set of circles—is musicalized as a looping rhythmic pattern. The number of notes in a pattern corresponds to the number of circles represented visually. These notes are distributed as evenly as possible over the steps of the pattern to form Euclidean rhythms as by Toussaint [2005]. The user adjusts the number of notes contained in a pattern by steering the visual object and collecting distinct objects depicting notes and rests. A change in a rhythmic pattern triggers a re-distribution of the notes. The change is not audible before the next onset in the new pattern is played, but it is immediately visible in the visual representation.

Finally, to make the application more personalized, the user initializes the software with another distinct type of input: a photo taken with the phone's built-in camera. A blurred version of the photo is used as the board on which the objects are visualized, and from which they draw their musical characteristics as described above.

The musical ground rules, namely tempo and scale, are also governed by the colors of the image. First, the image data is processed with a simple pipeline to get a single color representing the complete image. The tempo is then assigned to 110 ± 10 beats per minute according to the strength of the color (the sum of the saturation and brightness components). The scale, i.e., the overall tonality, is selected by straightforwardly mapping the hue, saturation, and brightness components to major, harmonic minor, and natural minor scales. Warm colors (such as bright, saturated yellows) are mapped to the major scale while cold colors (such as blues and saturated dark shades in general) are mapped to the harmonic minor, with the rest of the colors being mapped to the natural minor. In addition, the saturation controls the transposition (±1 half-step). Additionally, the color also affects polyphony of the improvisation mode described below; by default four voices are used, but for highly saturated colors with a single dominating RGB color component, the polyphony is reduced to three voices to achieve a slightly more focused sonic result.

Improvisation Mode. In the *improvisation mode*, the Musicreatures application enables the user to control the musical output more directly using gestures without visualization as an intermediary. The creative act in the improvisation mode resembles that of playing an instrument, whereas the visualized mode is analogous to conducting an ensemble.

The improvisation mode provides the user a means to play unpitched percussion instrument sounds and a polyphonic synthesizer and to control chord changes. The user controls these aspects with motion gestures. In order to keep the gestures simple, but at the same time prevent unsatisfactory chord progressions and dissonant voicings, the system imposes strong constraints on the chord changes and the harmonies. The chord progression is generated when the application is initialized with a background image, but the user controls whether the harmony advances from one chord of the progression to another. The user can initiate a chord change by moving the phone.

The polyphonic synthesizer is played whenever the phone is moved, and each of the percussion sounds is played when the device acceleration in a specific direction is greater than some threshold. The outputted percussion note onsets are quantized to ensure precise timing of these events. As discussed above, the repercussion of this approach is lag. The adverse effect of the delayed response is counterbalanced with direct, non-delayed control over the polyphonic synthesizer. However, instead of mapping gestures to onsets of fixed-duration note events, the motion of the device controls amplitude and filter envelopes. This gives the user a great amount of control over dynamics: rapid gestures can be used for creating note onsets and more relaxed ones for controlling the sustain. The pitch change timings are nevertheless quantized, which prevents the result from becoming sloppy. This combination allows the user input to result in immediate changes in timbre, reducing the effect of the lag caused by keeping some temporal events quantized.

Notes on Musicreatures. The Musicreatures application presents a challenging target for musicalization—one that involves controlling many distinct aspects of music while keeping the input gestures, and therefore the input data, simple and minimal. We next report briefly on the success of musicalization in Musicreatures, both from our subjective point of view and based on informal feedback from musicians and creative professionals. Readers are encouraged to try out the Musicreatures app for themselves; sample music (without the interactive component) is also available for listening in the Online Appendix.

The sonic output produced by Musicreatures is, according to our subjective preference, musical yet somewhat repetitive and limited. The music has a strong sense of rhythm and the overarching harmonic feel is consonant. However, there is a disjunction between the user's input motion gestures and the resulting music, especially in the mode where the user is in direct control of merely the visual groups. In the improvisational mode, the user does have a more intimate sense of control over the produced music in terms of dynamics and timing, but as concerns the compositional aspects of the music, the user can observe only very limited possibilities to affect the outcome.

Musicreatures has been used in a public improvisatory performance for a live audience at *The Science Forum* event in Helsinki, 10th January 2015. The ensemble consisted of the application (and its player, gesturing to produce music) and two professional musicians: a cellist and a bass clarinetist. A video recording of the improvisation session is publicly available[14]. The musicians deemed the result largely a success and a well-working act. The criticism concerned the ending of the performance, where the ensemble had to wait until the timers within the application stopped the musical patterns from being repeated. The musicians also had at times some difficulties in hearing what the application was playing because of the lack of control over dynamics in Musicreatures' standard mode.

The application was also demonstrated to professionals from creative industries as well as computational creativity researchers in a *One-Day Show, Tell and Imagine Workshop on Computational Creativity* at Queen Mary, University of London, 9th April 2015. Informal reaction from audience was very positive. In particular, the quality of the music was considered "really good" also among those who work with music professionally. This indicates success in the essential aspect of musicalization as producing music, not "just" sonification.

[14]https://youtu.be/DmQP5-AuDVQ

However, the relationship between the user's gestures and the produced music in the visually mediated standard mode was not obvious to many visitors.

The visually mediated musicalization of kinetic information enables the extra flexibilities discussed earlier in this section, but it also weakens the connection between the initial data and the resulting music. Furthermore, some musical parameters, e.g. the pitches, are controlled even less directly in order to decrease the monotony of the resulting music—instead of mapping a given color to a certain pitch, the colors control the maximum number of different pitches that can occur in a pattern. This approach increases the risk of making the causality too vague for the user. Adding more input gestures could enable more appropriate mappings between the data and the musical result, although care needs to be taken not to complicate the interaction.

Dividing the application into the two distinct modes increases the number of simple motion gestures that can be distinguished reliably at the expense of consistency between the modes. More sophisticated processing of motion data presents another approach—instead of relying only on very primitive motion gestures and changing their effects modally, more complex gestures could be used to enrich the input data of the musicalization. This approach provides a vastly different take on the gesture musicalization challenge and could be developed into a form that diminishes the need for the visual intermediary.

7. DISCUSSION AND CONCLUSIONS

We have defined data musicalization as automated composition of music based on given input data, an approach to perceptualize data in an interesting way, or even to create artistic artefacts. We identified defining characteristics of data musicalization, drawn from literature on sonification, automated composition of music and computational creativity (cf. Table I). We also gave seven key characteristics that help analyze and describe different data musicalization applications (cf. Table II). We then demonstrated data musicalization with several novel, implemented applications that go far beyond mere mechanical sonification of data. The application domains are diverse: sleep analysis, online chats, and a game-like application for mobile phones to produce music. Given the abundance of data in many fields and the many needs for understanding and using it, we believe that musicalization can be an attractive complementary approach to more traditional, objective data analysis or sonification techniques.

Based on our analysis of the field using the above mentioned concepts, musicalization has distinctive characteristics with respect to these fields. In comparison to sonification, musicalization is on one hand constrained to produce music; on the other hand, it has more freedom in how to relate audio to the input data, leaving room for machine creativity. A look at existing sonification applications confirms this observation. For example, the interactive mapping in LISTEN [Lodha et al. 1996; Wilson and Lodha 1996] and the direct mapping in CAITLIN [Vickers and Alty 1996; 2005] show that no musical knowledge was enforced to constrain the output. However, other sonifications, like the CLIMATE SYMPHONY [Quinn 2001] and LISTENING TO THE MIND LISTENING [Barrass et al. 2006a; Barrass et al. 2006b], have been presented in artistic venues. Therefore, the distinction between sonification and data musicalization is non-trivial. Already Vickers and Hogg [2006] argue that the difference between sonification (ars informatica) and music (ars musica) is slight. They suggest sonification and data musicalization are in the same continuum, but musicalization highlights the artistic goals whereas sonification's intent "is to create as pure a mapping from data to sound as possible". Furthermore, according to them, both sonification and data musicalization can have different levels of abstractness. We slightly disagree with this work: we do not consider direct mappings without musical models as musicalization, we rather assume that musicalization works on the metaphorical level and has low information capability (Table I). However, we agree with Vickers and Hogg [2006] that musicalization requires extensive focus on aesthetic aspects and maintain that sonification and data musicalization are different areas of research and practice.

In contrast to automated composition, musicalization is strongly data-inspired, unlike many composition applications that focus more on internal models and methods of the systems, or on supporting the user in composing music. Finally, when seen as a form of computational creativity, musicalization is characterized as generation of musical expressions for which the aesthetic is affected by the input data.

We have illustrated a large number of settings and techniques through various implemented applications (cf. Table II and Sections 3–6). These applications use different input data and have varying aims, from entertaining or ambient transfer of information to artistic expression. Some use a significant amount of hand-crafted musical knowledge, while some rely more on knowledge acquired through learning from musical corpora. The operating modes vary between batch, real-time and interactive operation, depending on the needs of the application.

The methods described in these examples serve as demonstrations of the multitude of possible approaches to musicalization as composition, and they could be improved in a number of ways to obtain better results. For instance, there is a large body of literature on automatic composition; here, we mention just a couple of promising avenues that depend on the application and its goals: production of a better global musical structure to improve the musicality of the results, higher perceptual relevance to improve the usefulness of musicalization in auditory displays, and more obvious response/causality to improve ease of interaction with musicalization software, where relevant.

For example, the interactive Musicreatures mobile application allows the user to take part in the music composition. Here, interactivity allows a symbiotic human-computer interaction in which both the user and the computer may adapt their behaviors based on each other's activity [Jacucci et al. 2014; Kantosalo and Toivonen 2016]. By contrast, the sleep musicalization examples give computers more creative responsibility. This can provide interesting opportunities to improve the composition of music through co-creative approaches, or, in the extreme even completely automatic ones.

One natural question that arises due to the unique characteristics of musicalization is how to evaluate the musicalization process and/or the artefacts it produces? We conducted a small user study on chat musicalization which demonstrated that users were able to relate a given text and its musicalization even without any instructions. Moreover, they liked the music more—possibly unconsciously—when it matched the text better. This supports the hypothesis that music can naturally reflect some characteristics of data, and that it thus can be a powerful way of representing data, albeit in a subjective way. However, it should be noted that such an evaluation was possible only because there was an explicit aim to convey information. This is not always the case in data musicalization, as seen in our other applications where the goal was more subjective—some form of pleasure or even self-expression. In such cases, less quantitative evaluation mechanisms may be appropriate.

Future Musicalization Applications. Particularly interesting for data musicalization seem to be applications with dynamic, rich, and especially personal data. Dynamic changes in data are naturally mapped to the temporal dimension of music; richness of data allows the data to control more nuances of the composition process. Most importantly, the uniqueness of automatically composed music highlights the particularity of the data and indirectly of its owners; real-time musicalization of dynamic data further emphasizes its transience and uniqueness. Promising application areas include social media, usage logs of individuals, and social networks and group interactions. Also, the physical environment and one's interaction with it are now sensed routinely, e.g. by accelerometers, positioning techniques, and light sensors in smart phones, potentially providing ample personal data for creative musicalization. Another promising application area for musicalization is multimodal interaction, e.g., enhancing user experience and enjoyment through use of multisensory media.

A specific area of future research is an examination of the impact of the musical component on the use of interactive or social applications. For example, in online chats, can we use musicalization to provide behavioral cues to participants, such as perceptualizing the tone of discussion or the use of offensive language? Even further, could the user's behavior be affected by suitable musicalization strategies, e.g., by making the user better aware of his tendency to use offensive language and thus helping him avoid using it?

Supplementary Materials

An Online Appendix at the journal web site contains sample music files produced with all the methods described in this article.

Samples are also available at

— http://sleepmusicalization.net/song.html
— http://mind.cs.byu.edu/music/sleepMusic/

Source code for some of the applications is available at

— https://github.com/Tulilaulu/Sleep-musicalization (perceptualization of sleep patterns)
— https://github.com/beddit/sleep-musicalization-web (the related web application)
— https://github.com/DiscoveryGroup/musicreatures (Musicreatures iPhone app)

REFERENCES

Florian Alt, Alireza S. Shirazi, Stefan Legien, Albrecht Schmidt, and Julian Mennenöh. 2010. Creating Meaningful Melodies from Text Messages. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Kirsty Beilharz, Bert Bongers, Andrew Johnston, and Sam Ferguson (Eds.). Sydney, Australia, 63–68.

Stephen Barrass, Mitchell Whitelaw, and Freya Bailes. 2006a. Listening to the mind listening: An analysis of sonification reviews, designs and Correspondences. *Leonardo Music Journal* 16 (2006), 13–19.

Stephen Barrass, Mitchell Whitelaw, and Guillaume Potard. 2006b. Listening to the mind listening. *Media International Australia incorporating Culture and Policy* 118, 1 (2006), 60–67.

Terri L. Bonebright and John H. Flowers. 2011. Evaluation of Auditory Display. In *The Sonification Handbook*, Thomas Hermann, Andy Hunt, and John G. Neuhoff (Eds.). Logos Publishing House, 111–144. http://sonification.de/handbook/download/TheSonificationHandbook-chapter6.pdf

Erin Bradner, Wendy A Kellogg, and Thomas Erickson. 1999. The adoption and use of "BABBLE": a field study of chat in the workplace. In *Proceedings of the Sixth European conference on Computer supported cooperative work*. 139–158. DOI:http://dx.doi.org/10.1007/978-94-011-4441-4{_}8

Roberto Bresin and Anders Friberg. 2011. Emotion rendering in music: range and characteristic values of seven musical variables. *Cortex; a journal devoted to the study of the nervous system and behavior* 47, 9 (10 2011), 1068–81. DOI:http://dx.doi.org/10.1016/j.cortex.2011.05.009

Amilcar Cardoso, Tony Veale, and Geraint A. Wiggins. 2009. Converging on the Divergent: The History (and Future) of the International Joint Workshops in Computational Creativity. *AI Magazine* 30 (2009), 15–22. Issue 3.

Simon Colton, John Charnley, and Alison Pease. 2011. Computational Creativity Theory: The FACE and IDEA Descriptive Models. In *Proceedings of the Second International Conference on Computational Creativity (ICCC)*.

Simon Colton and Geraint A. Wiggins. 2012. Computational creativity: The final frontier?. In *European Conference on Artificial Intelligence (ECAI)*. 21–26.

Reagan Curtis. 2004. Analyzing Students' Conversations In Chat Room Discussion Groups. *College Teaching* 52, 4 (10 2004), 143–149. DOI:http://dx.doi.org/10.3200/CTCH.52.4.143-149

Georg Essl and Michael Rohs. 2009. Interactivity for mobile music-making. *Organised Sound* 14, 02 (2009), 197–207.

Jose David Fernández and Francisco Vico. 2013. AI methods in algorithmic composition: a comprehensive survey. *Journal of Artificial Intelligence Research* 48 (2013), 513–582.

John H. Flowers, Dion C. Buhman, and Kimberly D. Turnage. 2005. Data Sonification from the Desktop: Should Sound Be Part of Standard Data Analysis Software? *ACM Trans. Appl. Percept.* 2, 4 (Oct. 2005), 467–472. DOI:http://dx.doi.org/10.1145/1101530.1101544

Justin Grimmer and Brandon M. Stewart. 2013. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis* 21, 3 (7 2013), 267–297. DOI:http://dx.doi.org/10.1093/pan/mps028

MH Hansen and B Rubin. 2001. Babble online: applying statistics and design to sonify the internet. In *Proceedings of the 2001 International Conference on Auditory Display*. 10–15. http://www.acoustics.hut.fi/icad2001/proceedings/papers/hansen.pdf

Mark Hansen and Ben Rubin. 2002. Listening post: Giving voice to online communication. In *Proceedings of the 2002 International Conference on Auditory Display*. 3–6.

Ewan Hill and Steven Goldfarb. 2016. *ATLAS data sonification: a new interface for musical expression and public interaction.* Technical Report ATL-OREACH-PROC-2016-005. CERN, Geneva. http://cds.cern.ch/record/2227055

C. Iber, S. Ancoli-Israel, A. Chesson, and S. F. Quan. 2007. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications.* American Academy of Sleep Medicine.

Giulio Jacucci, Anna Spagnolli, Jonathan Freeman, and Luciano Gamberini. 2014. Symbiotic Interaction: A Critical Definition and Comparison to other Human-Computer Paradigms. In *Symbiotic Interaction*, Giulio Jacucci, Luciano Gamberini, Jonathan Freeman, and Anna Spagnolli (Eds.). Lecture Notes in Computer Science, Vol. 8820. Springer International Publishing, 3–20. DOI:http://dx.doi.org/10.1007/978-3-319-13500-7_1

Daniel Johnson and Dan Ventura. 2014. Musical motif discovery in non-musical media. In *Proceedings of the Fifth International Conference on Computational Creativity (ICCC)*. 91–99.

PN Juslin, A Friberg, and R Bresin. 2002. Toward a computational model of expression in music performance: The GERM model. *Musicae Scientiae* 5, 1 suppl (2002), 63–122. http://doi.apa.org/?uid=2002-17591-004

P. N. Juslin. 2001. Communicating emotion in music performance: a review and a theoretical framework. In *Music and emotion: theory and research*, P. N. Juslin and J. A. Sloboda (Eds.). Oxford University Press, New York, 309–337.

Anna Kantosalo and Hannu Toivonen. 2016. Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity. In *The Seventh International Conference on Computational Creativity (ICCC)*. Paris, France, 77–84.

Helmut Kirchmeyer. 1968. On the historical constitution of a rationalistic music. *Die Reihe* 8 (1968), 11—24.

Daniel J. Levitin, Karon MacLean, Max Mathews, Lonny Chu, and Eric Jensen. 2000. The perception of cross-modal simultaneity (or "the Greenwich Observatory Problem" revisited). *AIP Conference Proceedings* 517, 1 (2000), 323–329. DOI:http://dx.doi.org/10.1063/1.1291270

Yin-tzu Lin, I-ting Liu, Jyh-shing Roger Jang, and Ja-ling Wu. 2015. Audio Musical Dice Game. *ACM Transactions on Multimedia Computing, Communications, and Applications* 11, 4 (6 2015), 1–24. DOI:http://dx.doi.org/10.1145/2710015

Suresh K. Lodha, Catherine M. Wilson, and Robert E. Sheehan. 1996. LISTEN: Sounding Uncertainty Visualization. In *Proceedings of the 7th Conference on Visualization '96 (VIS '96)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 189–ff. http://dl.acm.org/citation.cfm?id=244979.245053

T. M. Madhyastha and D. A. Reed. 1995. Data sonification: do you see what I hear? *IEEE Software* 12, 2 (Mar 1995), 45–56. DOI:http://dx.doi.org/10.1109/52.368264

Kristine Monteith, Tony Martinez, and Dan Ventura. 2010a. Automatic Generation of Music for Inducing Emotive Response. In *Proceedings of the First International Conference on Computational Creativity*. 140–149.

Kristine Monteith, Tony Martinez, and Dan Ventura. 2010b. Computational Modeling of Emotional Content in Music. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*. 2356–2361.

Keith Muscutt. 2007. Composing with algorithms: An interview with David Cope. *Computer Music Journal* 31, 3 (2007), 10—22.

Jieun Oh, Jorge Herrera, Nicholas J Bryan, Luke Dahl, and Ge Wang. 2010. Evolving the mobile phone orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. Sydney, Australia, 82–87.

J Paalasmaa, L Leppäkorpi, and M Partinen. 2011. Quantifying respiratory variation with force sensor measurements. In *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'11*. 3812—3815.

Joonas Paalasmaa, Hannu Toivonen, and Markku Partinen. 2015. Adaptive Heartbeat Modeling for Beat-to-beat Heart Rate Measurement in Ballistocardiograms. *IEEE Journal of Biomedical and Health Informatics* 19, 6 (2015), 1945–1952.

J. Paalasmaa, M. Waris, H. Toivonen, L. Leppäkorpi, and M. Partinen. 2012. Online monitoring of sleep at home. In *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'12*.

George Papadopoulos and Geraint Wiggins. 1999. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposim on Musical Creativity*. Edinburgh, UK, 110–117.

W. Gerrod Parrott. 2001. *Emotions in Social Psychology*. Psychology Press, Philadelphia.

James W Pennebaker, Roger J Booth, and Martha E Francis. 2007. Linguistic inquiry and word count: LIWC [Computer software]. *Austin, TX: liwc. net* (2007).

Z. Pousman and J. Stasko. 2006. A taxonomy of ambient information systems: four patterns of design. *Proceedings of the working conference on Advanced visual interfaces* (2006), 67–74. DOI:http://dx.doi.org/10.1145/1133265.1133277

Marty Quinn. 2001. Research set to music: The climate symphony and other sonifications of ice core, radar, DNA, seismic and solar wind data. In *Proceedings of the 7th International Conference on Auditory Display*.

C Roads. 1996. *The computer music tutorial*. The MIT Press.

Ian Simon, Dan Morris, and Sumit Basu. 2008. MySong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 725–734.

Robert Smith, Aaron Dennis, and Dan Ventura. 2012. Automatic Composition from Non-musical Inspiration Sources. In *Proceedings of the Third International Conference on Computational Creativity*. 160–164.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in short strength detection informal text. *Journal of the American Society for Information Science and Technology* 61 (2010), q. DOI:http://dx.doi.org/10.1002/asi.21416

Godfried Toussaint. 2005. The Euclidean Algorithm Generates Traditional Musical Rhythms. In *Renaissance Banff: Mathematics, Music, Art, Culture*, Reza Sarhangi and Robert V. Moody (Eds.). Bridges Conference, Southwestern College, Winfield, Kansas, 47–56. http://archive.bridgesmathart.org/2005/bridges2005-47.html

Takahiko Tsuchiya, Jason Freeman, and Lee W Lerner. 2015. Data-to-music API: Real-time data-agnostic sonification with musical structure models. In *The 21st International Conference on Auditory Display (ICAD–2015)*. Graz, Austria, 244–251.

Aurora Tulilaulu, Joonas Paalasmaa, Mikko Waris, and Hannu Toivonen. 2012. Sleep Musicalization: Automatic Music Composition from Sleep Measurements. In *Eleventh International Symposium on Intelligent Data Analysis (IDA) (LNCS)*, Vol. 7619. Helsinki, Finland, 392–403.

L. Turchet and R. Bresin. 2015. Effects of Interactive Sonification on Emotionally Expressive Walking Styles. *IEEE Transactions on Affective Computing* 6, 2 (April 2015), 152–164. DOI:http://dx.doi.org/10.1109/TAFFC.2015.2416724

L. Turchet and A. Rodà. 2017. Emotion Rendering in Auditory Simulations of Imagined Walking Styles. *IEEE Transactions on Affective Computing* 8, 2 (April 2017), 241–253. DOI:http://dx.doi.org/10.1109/TAFFC.2016.2520924

P Vickers and J.L Alty. 1996. CAITLIN: A musical program auralization tool to assist novice programmers with debugging. In *Proceedings of the International Conference on Auditory Display (ICAD '96)*. 17–24.

Paul Vickers and James L Alty. 2005. Musical program auralization: Empirical studies. *ACM Transactions on Applied Perception (TAP)* 2, 4 (2005), 477–489.

Paul Vickers and Bennett Hogg. 2006. Sonification Abstraite/Sonification Concr\ete: An'Aesthetic Perspective Space'for Classifying Auditory Displays in the Ars Musica Domain. In *Proceedings of the 12th International Conference on Auditory Display*.

Bruce N. Walker and Michael A. Nees. 2011. *Theory of Sonification*. 9–39 pages.

Ge Wang, Georg Essl, and Henri Penttinen. 2008. Do mobile phones dream of electric orchestras?. In *Proceedings of the International Computer Music Conference (ICMC)*. Belfast, United Kingdom.

Ge Wang, Georg Essl, Jeff Smith, Spencer Salazar, P Cook, Rob Hamilton, Rebecca Fiebrink, Jonathan Berger, David Zhu, Mattias Ljungstrom, and others. 2009. Smule = sonic media: An intersection of the mobile, musical, and social. In *Proceedings of the International Computer Music Conference (ICMC)*. Montreal, Canada, 283–286.

Catherine M Wilson and Suresh K Lodha. 1996. Listen: A data sonification toolkit. In *International Conference on Auditory Display*.