Since the first computers were used to make music the database has been an invisible partner in music literature. What is this database? Certainly, it is not a single database. I propose to understand the database as the concept of data organization for the purpose of optimizing computer performace. As a structure that lays beneath the surface of software programming, the database has been a focus of silent —but nonetheless present— activity in our computer music research comunity. I argue that by shedding some light on this inherent aspect of computers we can arrive at a clearer notion of how a database can sound. The premise is the following: computers cannot exist without databases. Thus, we can ask ourselves if all computer music is not database music.

There are overt and covert uses of the database, but the database is ubiquitous in all computer practices in general, and in computer music performance in particular. Three practices using computers and sound —Music Information Retrieval (MIR), sonification, and computer music— allow us to think database performance in a choreographic way. That is to say, the movement that we as 'computer musicians,' as 'sonificators,' and as music information 'retrievers' (in sum, as databasers), perform in the continuous dance of our practices.

0.1 mir

In MIR, the database is *in front* of the programmer, *next* to the computer. This practice combines Information Retrieval (IR) with Music Theory, and it has been present in academia for a while, most generally within Electrical Engineering departments. The objective of MIR is to obtain useful information from the analysis of sound signals. That is, MIR seeks to represent a complex signal with a small number of data points, thus defining a a navigable 'information space,' which is, quite literally, the discretized space of the database.

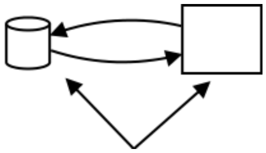


Figure 1: Diagram of database performance in MIR practices.
The database is visibly next to the computer, and the two bottom arrows indicate the intervention of the human operator.

For instance, out of sound file containing millions of samples, information space reduces these points to a database of few ‘descriptors’ that point to certain ‘features’ of the sound file. A descriptor is, in essence, a small amount of data that identifies other larger data. In this case, a feature descriptor relates to the values of a certain characteristics of the analyzed audio file, such as spectral centroid, brightness, flatness, etc.

Over the 18 years of the International Society for Music Information Retrieval (ISMIR) conference, more than thirty databases of this sort have been publicly created and released, as a means to classify millions of songs and musical genres. This type of database navigation has been used to perform automatic tasks such as categorization for recommendation systems (G. Tzanetakis & Cook, 2002; Dinuzzo, Pillonetto, & Nicolao, 2008; Poddar, Zangerle, & Yang, 2018), track separation or instrument recognition, and score transcriptions, among other uses (see below). A recent emphasis in open source database creation has gained momentum (Fonseca et al., 2017), such as the Freesound or Looperman.com (Looperman) databases, or Centre des Musiques Arabes et Mediterraneennes (CMAM)’s Telemeta, both collaborative database systems: the first two for general sound file sharing and classification, the latter for ethno-musicological purposes. Audio databases such as Freesound or Looperman have been growing exponentially, as well as their use within live performances and interactive systems (Correia, 2010). Automatic audio description and clustering among these databases automatic have improved greatly their usability (Xambo, Roma, Herrera, & Laney, 2012). Collins (2015) created open-source software implementing MIR techniques for navigation, analysis, and classification of the electronic music archive within UbuWeb.

Before sound and audio descriptor databases, however, music notation databases have been developed with a variety of file formats (See 0.0.7). Some examples of these notation databases can be the Polish folk song database in the Essen Associative Code (EsAC) format, the electronic library for musical scores MuseData, the Repertoire International des Sources Musicales (RISM) database, the *Kern Scores* database,[1] among others. In turn, these databases have been a fruitful area of exploration in Computational Musicology (Yolk, Wiering, & van Kranenburg, 2011), for which toolkits such as Massachusetts Institute of Technology (MIT)'s Music21 have been developed. Two examples of widely used libraries for audio analysis, classification, and synthesis are Music Analysis, Retrieval and Synthesis for Audio Signals (Marsyas) (George Tzanetakis & Cook, 2000) and the Essentia (Bogdanov et al., 2013). For a more general overview of MIR software, see (ibid.). The different applications of databases are endless and so varied that would extend the scope of this study.[2] Some specific uses that MIR has given to databases have been:

- for audio classification and clustering (Yang, 2001; Homburg, Mierswa, Möller, Morik, & Wurst, 2005; Queiroz & Yoshimura, 2018)
- for genre recognition and classification (G. Tzanetakis & Cook, 2002; Xu, Zang, & Yang, 2005; Jr., Koerich, & Kaestner, 2008; Sanden, Befus, & Zahng, 2010; Dehkordi & Banitalebi-Dehkordi, 2018; X. Wang & Haque, 2017; Mitra & Saha, 2014; Correa, Saito, & Costa, 2010; Dinuzzo et al., 2008)
- to describe performance expression (Hashida, Matsui, & Katayose, 2008; Hashida, Nakamura, & Katayose, 2017; Hashida, Nakamura, & Katayose, 2018)
- for emotion recognition and color associations in the listener (Pesek et al., 2014)
- for multimodal mood prediction (Delbouys, Hennequin, Piccoli, Royo-Letelier, & Moussallam, 2018; Hu & Yang, 2014; Corona & O'Mahony, 2015)

[1] http://kern.ccarh.org/

[2] For example, consider the Musical Instrument Museums Online (MIMO) database, a project dedicated to the cataloguing of musical instruments, and how it was used for the statistical tracking of the evolution of the violin based on pattern recognition of its shapes (Peron, Rodrigues, & Costa, 2018)

- for multi-instrument recognition (Humphrey, Durand, & McFee, 2018)
- for the evaluation of multiple-source fundamental frequency estimation algorithms (Yeh, Bogaards, & Röbel, 2007)
- for contextual music listening pattern detection using social media (Hauger, Schedl, Kosir, & Tkalcic, 2013)
- for melody (Karydis, Nanopoulos, Papadopoulos, Cambouropoulos, & Manolopoulos, 2007; Bittner et al., 2014) or singing voice (Stoller, Ewert, & Dixon, 2017) extraction
- for structural analysis (J. B. L. Smith, Burgoyne, Fujinaga, Roure, & Downie, 2011)
- for schenkerian analysis (Kirlin, 2014)
- for harmonic analysis (Devaney, Arthur, Condit-Schultz, & Nisula, 2015)
- for melodic similarity (Haus & Pinto, 2005)
- for forensic analysis as a complement of video analysis (Serizel, Bisot, Essid, & Richard, 2016)
- for the evaluation of tempo estimation and key detection algorithms (Knees et al., 2015)
- for tonal music analysis using Generative Theory of Tonal Music (GTTM) (Hamanaka, Hirata, & Tojo, 2014)
- for counterpoint analysis (Antila & Cumming, 2014)
- to train models for phoneme detection (Proutskova, Rhodes, Wiggins, & Crawford, 2012) and music source separation (Miron & Janer, 2017)
- for training and evaluating chord transcription algorithms (Eremenko, Demirel, Bozkurt, & Serra, 2018)
- for training querying methods (Cartwright & Pardo, 2012; Brzezinski-Spiczak, Dobosz, Lis, & Pintal, 2013; Nagavi & Bhajantri, 2014; Nagavi & Bhajantri, 2013; Melucci & Orio, 1999)
- for adversarial audio synthesis (Donahue, McAuley, & Puckette, 2018)
- for orchestration (Crestel, Esling, Heng, & McAdams, 2017)
- for modeling carnatic rhythm generation (Guedes, Trochidis, & Anantapadmanabhan, 2018)
- to create digital libraries (Dunn, 2000)
- to store music notation (Good, 2000)

For further reference, the following citations point to different audio databases which have been created over the years: Goto, Hashiguchi, Nishimura, and Oka (2002), Goto, Hashiguchi, Nishimura, and Oka (2003), Wüst and Celma (2004), Maxwell and Eigenfeldt (2008), Bertin-Mahieux, Ellis, Whitman, and Lamere (2011), Karaosmanoglu (2012), Jaimovich, Ortiz, Coghlan, and Knapp (2012), Mital and Grierson (2013), Bortz, Jaimovich, and Knapp (2015), Jaimovich and Knapp (2015), Nort, Jarvis, and Palumbo (2016), Defferrard, Benzi, Vandergheynst, and Bresson (2017), Vigliensoni and Fujinaga (2017), Meseguer-Brocal, Cohen-Hadria, and Peeters (2018), Donahue, Mao, and McAuley (2018), Xi, Bittner, Pauwels, Ye, and Bello (2018), Wilkins, Seetharaman, Wahl, and Pardo (2018).

0.2 sonification

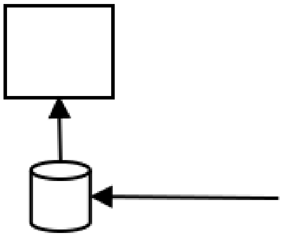


Figure 2: Diagram of database performance in sonification practices.
The database is visibly below the computer, and it feeds the computer from an external source represented by the right-most arrow.

The database is the ground floor of sonification. The sonified data is very likely to be digital,[3] which means that data needs to be stored in a structured way for fast access by computers, and the role of the sonifier is to acoustically translate the database's inner relationships (Walker & Nees, 2011, p. 9).

According to Walker and Nees (ibid.) there are three types of sonification:

[3] There are cases where sonification is entirely analog, such as the first sonification tool ever created: the Geiger counter

event-based, model-based, and continuous. I see these types of sonification as ways of performing a database. Continuous sonification (audification) consists of directly translating waveforms of periodic data into sound, that is, reading non-audio data as if it were audio data (ibid., p. 17). Model-based sonification consists of distributing data points in such a way that enables data exploration. Generally, these models are interactive interfaces with which users navigate the database to find relationships (ibid., p. 17). Event-based (parameter mapping) sonification is aimed at representing changes in a database as acoustic saliences or tendencies. In this sense, dimensions of the data need to be translated (mapped) into acoustic parameters (frequency, periodicity, density, etc.), so as to listen how the generated sound behaves over time and interpret these changes within the database (ibid., p. 16).

Sonification depends on databases, on the interaction between databases, and on their traversing, but also on the human body's perceptual limits. In sonification, the data comes first, and it needs to be pre-processed so that it can be adapted to the sound synthesis engines of choice. Sonification is a subset of auditory display techniques, and it belongs to the broader scope of information systems and visualization practices (ibid., p. 10). Therefore, since sonification belongs to the process of information, as a practice it has taken into account the auditory system's ability to extract biologically relevant information from the complex acoustic world (Carlile, 2011). What this emphasis on sound perception and cognition abides to, however, is the fact that there is no one-to-one correspondence between sound parameters (frequency, amplitude, spectral content) and how these are perceived (pitch, loudness, timbre). Therefore, the success of a sonification is the result of the play between, on the one hand a rigid link between data and sound, and on the other, the perceived acoustic relations. From this interplay of relations is how information can be obtained from data. In other words, in sonification practices there is no communication unless the data has been acoustically shaped, and perceived as information (*in*-formed) by the listener.

In what follows, I present some instances of sonification practices as described by their authors.

0.2.1 Parameter mapping

DOW Rossiter and Ng (1996) sonified the Dow Jones financial stock market data with Csound. Since the Csound program depends on two separate files (orchestra and score), they implemented another program to control the data flow. Within this second program, the Csound score was automatically generated based on a 'configuration' file which was used to map the 'data file' holding the stock market data, as it was read in separate window frames into the Csound-formatted score.[4]

Medical Images Cádiz et al. (2015) proposed a sonification approach based on statistical descriptors of Region of Interest (ROI) selected from medical images. In their study, they focused on enhancing breast cancer symptom detection in mammograms by mapping statistical descriptors, such as mean, minimum, maximum, standard deviation, kurtosis, skewness, among others, to different synthesis techniques in various ways. They then surveyed the usefulness and pleasantness of the sonifications to different subjects in order to better adjust the technique to the task. What is novel of their approach is on the creative use of statistical curves obtained from pixel distributions within computer music techniques.

0.2.2 Model-based sonification

Space One example of model-based sonification is the *Data Listening Space* installation by the QCD-Audio project at the Institute of Electronic Music and Acoustics (IEM) of the University of Music and Performing Arts in Graz (Vogt, Pirro, Rumori, & Hoeldrich, 2012). Within this installation, they proposed a three dimensional, naviga-

[4] Other examples of stock market sonification include Ciardi's set of tools for downloading and sonifying real-time data, see Ciardi (2004); and Ian Whalley's research on telematic performance, see Whalley (2014)

ble space holding a Monte Carlo simulation of the theory of Quantuum Electrodynamics (QED). Within this QED *lattice*, a walking participant holding sensors —x, y, and z coordinates— could explore the simulated data by way of sonification.

0.2.3 Artistic sonification

Wolves Klein (1998) composed a piece called “The Wolves of Bays Mountain”, using a set of recordings she took along the Bays Mountain Park in Kingsport, Tennessee, for a period of six months. In this period she researched the sonic activity of a pack of wolves, and in her recordings she achieved a level of intimacy with the pack that translated into the recordings, and resulted in a strong animal rights activism (Klein, 2017). Therefore, her compositional choice was to treat the sound file in a non-destructive and non-intrusive way: “for the composition I used the Csound computer music language. All of the sounds came from the recordings, in unaltered or slightly modified form as the source material in musical settings and transitions” (Klein, 1998). Thus, by analyzing spectral contours of extremely precise frequency bandwidths of the data and resynthesizing into the soundscape in almost unnoticeable ways, she sonified a space in between the wolves. This space invites the listener into a space of action, and to reflect on human activity itself and how it always returns to resonate with the wolves.

Selva Barrett (2000a) composed an electroacoustic work called “Viva La Selva” (Barrett, 2000b) using 14-hour long recordings taken with an array of four microphones from a a biological field station called *La Suerte* in Costa Rica. From these recordings, she extracted location (by difference in arrival time) and timestamps (by manual logging) of different animal sounds, and long-term energy distribution in various frequency bands, to describe various environmental sounds such as airplanes, wind, insects, etc. While the spatio-temporal data of the animal sounds was used for sound spatialization of sounds within the electroacoustic work, the long-term energy distribution was scaled down to 20 minutes so as to constitute the form of the piece.

Ocean Sturm (2002) sonified ocean wave conditions of the USA Pacific coast obtained by the Coastal Data Information Program (CDIP) since 1975. The database until 2002 contained over 50 Gigabyte (GB) of spectral and directional content of the wave-driven motions at the location of the sensing buoys. By scaling to hearable range and then performing an Inverse Fourier Transform (IFT) of the data, Sturm composed a piece called *Pacific Pulse*, on which frequency sweeps indicate storms beginnings (rising) and endings (falling).

Molecules Falk Morawitz (2016) composed *Spin Dynamics* using molecular sonification by two audification processes (direct audification and via a straightforward additive synthesis process) applied to the Human Metabolome Database (HMDB), a database holding Nuclear magnetic resonance (NMR) spectroscopies of molecules.

Gender Distribution Emma Frid (2017) derived a database of gender distribution by applying the python module `genderize` to author names in three main computer music conference proceedings databases: International Computer Music Conference (ICMC), New Interfaces for Musical Expression (NIME), and Sound and Music Computing Conference (SMC). By assigning polar frequency ranges for each group (male and female), her sonification emphasizes the significant inequality of gender in the resulting acoustic stream segregation into male background (continuous drone-like sound) and female foreground (fewer and sparser sounds). Her conclusion, therefore, is that “there is a need for analysis of the existing environments and social relations that surround music technology and computer music. If we identify the challenges that women are facing in our research community, we will be able to create more initiatives towards changing practices” (ibid., p. 238).

0.2.4 Sonification Installations

IP-based soundscape Ballora, Panulla, Gourley, and Hall (2010) sonified a database of Hypertext Transfer Protocol (HTTP) requests at Penn State's Center for Network-Centric Cognition and Information Fusion (NC2IF). This database contained entries with four fields such as timestamp, location (latitude-longitude), Internet Protocol (IP) address, and response type. Using parameter mapping, Ballora controlled rhythm and spatialization with the first two, and pitch and timbre with IP data. However, the latter ranged from the more concrete (IP to frequency) to the more abstract (IP as formant and high-pass filters for brown noise), thus resulting in a soundscape with different but simultaneous sonifications of the data. This multi-layered approach to sonification stems from his PhD dissertation on cardiac rate sonification (Ballora, 2000).

Earthquakes Lindborg (2017) sonified real-time earthquake data as a sound sculpture. Within "Pacific Bell Tower, a sculptural sound installation for live sonification of earthquake data", he used data from the Incorporated Research Institutions for Seismology (IRIS) Data Services, which transmits seismographic data packets updated every thirty minutes from multiple observation sites. He spatialized this data using coordinates of the events and using a four-speaker array located at the center of the gallery space, and mapped the rest of the data to Frequency Modulation (FM) synthesis parameters.

GPU-based waveforms Schlei and Yoshikane (2016) proposed a novel way to generate waveforms by populating an array using vertex data obtained from the Graphics Processing Unit (GPU). In order to carry this out, they used the Metal API[5], and intervened on the processing pipeline to output Central Processing Unit (CPU) accessible data. The audio engine running on the CPU was able to interpret as waveforms the

[5] Apple's built-in framework to interface with the GPU. See https://developer.apple.com/documentation/metal

values of the vertex and fragment shaders, thus sonifying the position data related to a rendered shape and the pixel values respective to its display. Therefore, they obtained simultaneous visualization and audification of the rendered three dimensional shape. In their installation *The Things of Shapes*[6], they used the generated waveforms as a database, composing each waveform together with their visual generators as a collage.

Uncanny Faces Simonelli, Delgadino, and Halac (2017) designed *Hally*, an installation based on face tracking and real-time sonification of spectral features present in both pixel information containing the face, and the x and y coordinates of the moving data points of the face mesh used for tracking. Furthermore, by video-based audio convolution, *Hally* aims to simulate a theory of perception based on IFT (Connes, 2012). Parting from previous work by Thiebaut, Bello, and Schwarz (2007) on simultaneous sonification and visualization, *Hally* explores the role of both sound and image in the definition of the self, by immersing the participant in an uncanny spectrality (Cámara Halac, 2018). Among other sounds, there is one drone-like sound produced throughout the installation that comes out of a set of programming decisions which bring into surface certain aspects of the software used in the piece. The within Pure Data is scheduled in blocks of samples that the user can set. A certain combination of block size, overlap, and sample rate results in a frequency, in this case, of about 43 Hz (Simonelli et al., 2017, p. 3). In fact, these combinations around block size result in a set of octave-spaced pitches that are tethered to Pure Data's only possible block sizes (2^n, at least in audio computations). In this sense, when making these pitches audible, the listener can recognize a certain 'sound' proper to the internal memory management and computation rate of the software. In this installation, it is the human participant that enters as a filter to the spectrum of this 'sound' by the position of their faces on the camera sensing area. Thus, participants change this 'sound' with the unique ways in which they behave in front of the sensor.

[6] https://vimeo.com/167646306

0.2.5 Sonification Software

SonArt Originally intended for sonification purposes, SonART (Ben-Tal, Berger, Cook, Daniels, & Scavone, 2002) was an open-source platform that enabled users to map parameters to sound synthesis, and later (W. Yeo, Berger, Lee, & Zune, 2004) to obtain cross-correlated image and sound synthesis. In other words, users were able to easily translate a database into sound parameters, or image and sound data into one another. The program acted in a modular way, that is, it was networked with other software via Open Sound Control (OSC) connections. This software enabled W. S. Yeo and Berger (2005) to generate novel image sonifications, by combining two methods of sonification into one interface: sonified data in a fixed, non-modifiable order (*scanning*) and sonified selected data points (*probing*).

DataPlayer In his Computer Aided Data Driven Composition (CADDC) environment called *DataPlayer* programmed as a standalone MAX/MSP application, Nardelli (2015) sonified data from the Automatic Flow for Materials Discovery (AFLOWLIB). His sonification intent was aimed towards data navigation by means of a unique mapping that would convey an overall trend (a gist) of each material compound. Furthermore, this environment allowed for artistic remixing and exploration of the sonification procedures, simultaneously touching on the scientific and the artistic uses of the environment.

madBPM Fox, Stewart, and Hamilton (2017) devised madBPM, a data-ingestion engine suitable for database perceptualization, that is, sonification and visualization. This modular C++ software platform enables data loading from Comma Separated Values (CSV) files, multiple mapping via tagging, several traversing algorithms and units, and networked connectivity to SuperCollider for sound and OpenFrameworks (OFX) for visual output. Their approach is innovative since they provide features for database behaviors. By 'behavior' they mean ways of structuring, traversing and perceptualizing the database. These behaviors define the dual purpose of the software: finding relationships among the inputted data and interpreting them artistically. Furthermore, users can structure and re-structure potentially any type of data set (ibid., p. 504). However, in order to design new behavior objects the user needs to implement them in the source code and compile them. Thus, besides real-time data streaming and networking functionality, in their future work the authors aim at designing a Domain Specific Language (DSL) that would enable extending the functionality of these behaviors in real-time.

For further sonification software, see Sonifying Data (SonData) and the following references: Wilson and Lodha (1996), Pauletto and Hunt (2004), Lodha, Beahan, Joseph, and Zane-ulman (1998), Beilharz and Ferguson (2009), Hildebrandt, Hermann, and Rinderle-Ma (2014), Worrall, Bylstra, Barrass, and Dean (2007), Walker and Cothran (2003), Vicinanza (2006)

0.3 computer music

Computer music software is computer music's playground. Composing and programming blend into different forms of play that can be understood by a closer look at the playground's design. A key aspect of software design is delimiting constraints to data structures. The first choice is generally the programming language, after which the database tree unfolds its way up to the leaves. Among these leaves is where computer music programs reside. At this level of 'leaves' software users are certainly aware that there is a 'tree' in front of them. However, their awareness does not necessarily extend to the branches, trunk, or roots of the tree. There is endless music that can be made with leaves just as it can with paper. However, neither music quantity nor music quality are the point here. My argument is that working with data structures changes how we think and perform music making. I claim that composers using these leaves of computer music software are working indirectly with data structures, and unless they engage with programming, they remain unaware of data structures and their constraints. ‘Indirectly,’ because the twigs and branches connect the leaf to the trunk, but these connections become invisible to the non-programmer composer *by design*. Like a phantom limb of the tree, the database remains invisibly *behind*. In this section, I present different approaches from composers and programmers that show how music concepts change with the presence and performance of the database. By database performance I mean neither the quality of musical output, nor the dexterity of the programming activity. Database performance in music composition is the activity of the databaser: databasing to make music.

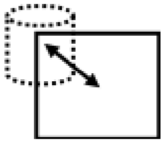


Figure 3: Diagram of database performance in computer music practices. The database is invisibly behind the computer, within the softwares used to create musical works.

0.3.1 Hierarchical environments

> One of the most important aspects in the design of any computer system is determining the basic data types and structures to be used. . . we have been guided by our projection of the interaction between the tool which we are developing, and the composer. (Buxton, Reeves, Baecker, & Mezei, 1978, p. 119)

Reducing cognitive burden In William Buxton’s survey of computer music practices (Buxton, 1977; Buxton, Reeves, et al., 1978; Buxton, Patel, Reeves, & Baecker, 1980), he distinguished between *composing programs* and *computer aided composition*, arguing that they had both failed as software, the former on account of their personalization and formalization, and the latter on their lack of interactivity. On his later interdisciplinary venture called Structured Sound Synthesis Project (SSSP), he focused on Human Computer Interaction (HCI) —a field in its very early stages in 1978—[7]. Buxton's concern throughout his work on SSSP was to address the"problems and benefits arising from the use of computers in musical composition" (Buxton, Fedorkow, et al., 1978, p. 472). His solution to the problems was to reduce the cognitive burden of the composer, who "should simply not have to memorize a large number of commands, the sequence in which they may be called, or the order in which their arguments must be specified" (ibid., p. 474). He argued that reducing the amount of information given to composers helped them focus on music making. Therefore, in SSSP, the composer's action was reduced to four main selection tasks: timbres, pitch-time structure, orchestration, and playback. Timbres were assigned by defining waveforms for the table lookup oscillators, and pitch-time structure consisted on pitches and rhythms on a score-like Graphical User Interface (GUI) program called SCRIVA (Buxton, 2016a). Orchestration consisted in placing the previously chosen timbres on the score, and playback meant running the score or parts of it. With this simple but very concise structure, stemming from a somewhat dated programming philosophy in relation to audio software, Buxton delimited the scope of action of the composer

A Hierarchical Representation Buxton, Fedorkow, et al. (1978) based their research on differing approaches to composition: Iannis Xenakis's score-as-entity approach in his 1971 *Formalized Music*, an unpublished 1975 manuscript by Barry Vercoe at MIT studio for Experimental Music, where Buxton found a note-by-note approach, and Barry Truax's computer music systems (Truax, 1973) which was, for Buxton, located somewhere in between the first two but did not provide a solution for "the problem of dealing with the different structural levels of composition —from note to score" (Buxton, Reeves, et al., 1978, p. 120) (See 0.0.6). Buxton, however, condensed these different approaches into what he called a "chunk-by-chunk" composition, where a 'chunk' represented anything from a single note to an entire score, and thus reframed

[7] William Buxton is now considered a pioneer in HCI, and he is now a major figure in the Microsoft Research department.

the question of a compositional approach as one of scale. For Buxton, "the key to allowing this 'chunk-by-chunk' addressing lies in our second observation: that the discussion of structural 'levels' immediately suggests a hierarchical internal representation of scores" (ibid., p. 120). That is to say, his solution for the scalability problem relied on a hierarchical representation of scores.

In Buxton's SSSP, the hierarchical design depended on a data structure called *symbol table*, which he subsequently divided into two objects called `score` and `Mevent` (musical events). The `score` structure had a series of global fields (variables) together with pointers to the first (head) and last (tail) `Mevents`. In turn, `Mevents` had local fields for each event together with pointers to the next and previous `Mevents`, so as to keep an ordered sequence (See **??**) and enable temporal traversing of the tree. In turn, `Mevents` could have two different types: `MUSICAL_NOTE` and `Mscore`, the former relating to terminal nodes editable by the user —what he referred to as 'leaves' of the tree structure—, and the latter consisting of nested `score` objects that added recursivity to the structure. Buxton's model was thus hierarchic (a tree structure) implemented in nested and doubly-linked symbol tables.

Buxton, Reeves, et al. (ibid.) gave a detailed exposition of the data structures and their functionality. Buxton's general purpose in his HCI philosophy was to make the software work in such a way that it became invisible or transparent to the user. This is also known as a black-box approach. His innovations in this and other projects have had enormous resonances in computer science, and the concept of reducing cognitive burden of the user has developed as a standard of HCI (Buxton, 2016b).

Black-boxing Media theorist Vílem Flusser (2011) proposed the term 'envision' to describe a person's power to visualize beyond the surface of the image, and to bring the technical image into a concrete state of experience. The 'image,' in Flusser's case is the television screen in its abstract state of "electrons in a cathode ray tube." Therefore, he argues, "if we are asking about the power to envision, we must let the black box remain

—cybernetically— black" (ibid., p. 35). By seeing past the abstract quality of media we bring an image into experience. The black box allows envisioning to take place. In a similar way, by seeing past the hidden complexities of the software, composers are able to create music with unrestrained imagination. However, as I have shown before, Hansen makes a divergent point claiming that the virtuality inherent in the body is the creative potential of image *in-formation* (See **??**).

Understanding the process of information as the experience of technical images, it follows that virtuality and envisioning can be considered complementary. On one hand, there is the technical device, whose multidimensionality is as complex as it is hidden from the envisioner. On the other, the human body with its capacity to create and embody. Flusser's point is, however, paradoxical: "The envisioner's superficiality, to which the apparatus has *condemned* him and for which the apparatus has *freed* him, unleashes a wholly unanticipated power of invention" [emphasis added] (ibid., p. 37). Therefore, the black-box is what condemns and frees the envisioner to a state of superficiality. However, Flusser continues, "envisioners press buttons to in-form, in the strictest sense of that word, namely, to make something improbable out of possibilities" (ibid., p. 37). In other words, Flusser justifies the invisibility of the technological device in favor of its most useful consequence, that is, its ability to make the user create something "out of possibilities." Composers, therefore, are often given these possibilities to create, at the cost of a restricted creation space.

Generality and Portability

> Music data structures must be general enough so that as many styles of music as possible may be represented. This implies that the data structures (or the application's interface to them) should not enforce a musical model (such as equal temperament) that is inappropriate for the musical task at hand. (Free, 1987, p. 318)

The SSSP lasted until 1982 due to lack of funding, and in the mid-1980s its research re-emerged with the work of Free and Vytas (1986), Free (1987), Free and Vytas (1988), under Helicon Systems' Computer Assisted Music Project (CAMP). Free's programming philosophy called for generality, portability, and simplicity. Due to SSSP's many hardware dependencies, the code had to be completely re-written (Free & Vytas, 1986). A crucial aspect of Free's programming concerns was portability (See **??**), which moved him to create higher levels of software abstractions, so that software continued to live on in newer hardware. Free also developed SCRIVA, SSSP's GUI program into extensible data structures for music notation arguing that software had to be general enough so that composers could work in multiple styles. The larger implication in Free's argument is that enforcing musical concepts in data structures limits the style that the program can achieve. Therefore, if the program fails to provide a certain level of generic functionality, the composer's output will be modelled by the data structure. On the one hand, it can be argued that this implication is simultaneously overestimating the agency of the database and underestimating that of the composer. In any case, the database works for the composer by taking care of the more tedious task. The cost of this, nonetheless, is that by working for the composer, the database guides the composer through certain paths while hiding other paths.

Simplification Hardware-independence led Free to imagine a general purpose, or *vanilla* synthesizer, with which students in "a music lab with multiple users on a networked computer system" (Free & Vytas, 1988, p. 127) could seamlessly use the timbre world offered by various synthesizers made by different manufacturers. Free created a database that enabled simultaneous interaction among different types of hardware. The *Music Configuration Database* consisted of an intermediate program between the physical Musical Instrument Digital Interface (MIDI) input devices (such as the Yamaha DX7 or Casio CZ101), and the computers in the network, so that "rather than have the user tediously specify the MIDI device properties for each synthesizer" (ibid., p. 133) (channel management, control mapping, etc), these processes were handled by an intermediary database. Free's approach, in comparison to Buxton's, was not entirely black-boxed, since the database was open to modification by a specific set of commands provided to the user. The user could edit the database with a library of database access subroutines such as open/close, create/delete items, querying fields/keys, and loading/storing property items. With this library, Free simultaneously simplified user's interaction and reduced the "chance of corrupting the database" (ibid., p. 137).

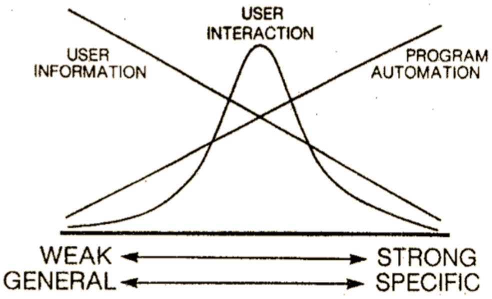


Fig. 1. Variation of user information, program automation, and user interaction as a function of the generality and strength of a system.

Figure 4: Generality vs. Strength
Barry Truax' "Inverse Relation Between Generality and Strength" (Truax, 1980, p. 51). Another version of this graph can be found in (Laske & Tabor, 1999, p. 38).

Balance

> ... all computer music systems both *explicitly and implicitly embody a model of the musical processes that may be inferred from the program and data structure of the system*, and from the behavior of user working with the system. The inference of this model is independent of whether the system designer(s) claim that the system reflects such a model, or is simply a tool. [emphasis added] (Truax, 1976, pp. 230–231)

Truax (1973), Truax (1976), Truax (1980), Emmerson (1986, Chapter 8) often compared grammatical structures of natural language to the structures of computer music systems, claiming that in both cases one can find certain constraints and facilitations for thought (Emmerson, 1986, p. 156). Arguing for balance between generality of applicability and strength of embedded knowledge within models for computer music systems (See Figure 5), he writes:

> In a computer music system, the grouping of data into larger units such as a sound-object, event, gesture, distribution, texture, or layer may have a profound effect on the composer's process of organization. The challenge for the software designer is how to provide powerful controls for such interrelated sets of data, how to make intelligent correlations between parameters, and how to make such data groupings *flexible according to context*. [emphasis added] (ibid., p. 157)

Truax's notion of balance speaks of a 'meeting halfway' between the system and the user regarding the programmer's capability to embed a more complex conception of hierarchy in the system. What provides this balance is a certain flexibility among data structures which would enable them to adapt to the different hierarchical contexts with which music is understood. That is to say, since data structures can embody models of musical processes, they have an effect on the composer's overall performance of the database, and by extension, on the resulting music.

0.3.2 Music Notation Software

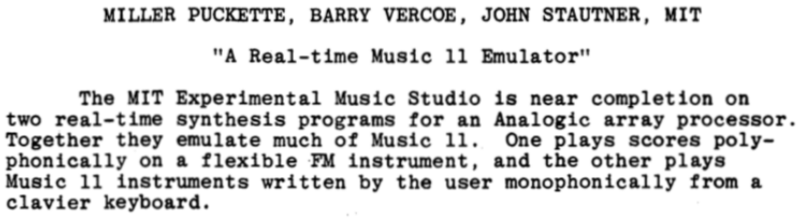
Music representation has occupied an important area of research within the programming community. Formats and specifications such as MIDI, MusicXML (MusicXML), the Humdrum `**kern` data format (Sapp, 2005), GUIDO Music Notation Format (GUIDO), to name a few, have appeared over the years in conjunction with music engraving software. An extensive guide on musical representations can be seen in Selfridge-Field (1997a). In this section, I point to certain aspects of music notation software development that reveal different approaches towards data structures, and the possibilities that arise henceforth.

DARMS and SCORE Two major programs were developed during the 1960s and 1970s: Stefan Bauer-Mengelberg's Digital Alternate Representation of Musical Scores (DARMS) project for music engraving which started in 1963 (Brinkman, 1983; Erickson, 1975), and Leland Smith's SCORE (L. Smith, 1972). Both of these programs worked first in mainframe computers and were used for music printing and publishing. At first, SCORE's character scanner was designed to interpret complex musical input into MUSIC-V output, thus acting as an link between music notation and computer music synthesis. However, with the appearance of vector graphics in the 1970s it shifted solely to music printing. With the appearance of the PostScript format in the 1980s, it became commercially available thus becoming one of the earliest music engraving softwares still in use today by major publishing houses (Selfridge-Field, 1997b).

From Staves to Speakers Other programming approaches stemming from DARMS and SCORE were developed during the 1980s. Clements (1980) joined together the DARMS data structures with those used in MUSIC-V in a first attempt to obtain sonic feedback out of a notation system. Clements' attempt was nonetheless overshadowed by SCORE's success. Later, Dydo (1987) worked on an interface to the DARMS language called the *Note Processor*, which became one of the earliest commercially available music notation systems. Dydo's data structures, however, were not publicly released when he presented his software at the ICMC in 1987. He later released it commercially in the early 1990s at a significantly lower price than other notation software such as *Finale* which is still available today by MakeMusic, Inc. (Skinner, 1990a; Skinner, 1990b). Brinkman (1981) modeled the SCORE input format into *Score-11*, adapting it to Barry Vercoe's MUSIC-11. Written in Pascal, *Score-11* used circular linked lists traversed by an interpreter to produce MUSIC-11-formatted output. The user creates a text file with blocks dedicated to individual instruments and specifies parameters such as rhythm, pitch, movement (glissandi, crescendo), amplitude, etc. These parameters are then re-formatted to fit the less musically-oriented notation of the MUSIC-N programs. Brinkman argued that such a software would result in faster and less arduous performance on the composer's end: "a crescendo over several hundred very short notes requires several hundred different amplitude values representing the increasing volume. *Typing in several hundred note statements each with a* *slightly larger amplitude number would take forever*. If the computer could be instructed to gradually increase the amplitude value over twenty seconds then *life would be much simpler*" [emphasis added] (Brinkman, 1982). Brinkman emphasized on the program's extensibility by users, inspiring Mikel Kuehn's recent *nGen* program (McCurdy, Heintz, Joaquin, & Knevel, 2015), a version of Brinkman's program for the current Csound. Brinkman (1983) later designed an interpreter for the DARMS language, which became useful for obtaining computable data structures for automated music analysis (Brinkman, 1984). Another approach to music notation was carried out at Center for Computer Research in Music and Acoustics (CCRMA), when G. Diener (1988), G. Diener (1989) devised a "pure structure" devoted to the "hierarchical organization of musical objects into musical scores:" the *TTree* (G. Diener, 1988, p. 184). Stemming from his PhD research on formal languages in music theory (G. Diener, 1985), this data structure was based in the hierarchic structures of the SSSP project. The change Diener introduced to these structures was their capability of sustaining links between not only the previous and the next data records, but to the 'parent' or 'child' data records to which it was related. This is known as 'inheritance,' and it enabled "any event in the [structure] to communicate with any other event" (G. Diener, 1988, p. 188). While Diener implemented this data structure in the object-oriented programming language Smalltalk, he later developed it into *Nutation* (G. R. Diener, 1992), a visual programming environment for music notation. *Nutation* was written in Objective-C, and it combined the previously developed *TTree* structure with glyphs and a music synthesis toolkit called *Music Kit* that the NeXT computer provided. This resulted in an extremely malleable Computer Aided Composition (CAC) environment, which enabled fast manipulation and sonic feedback at the cost of limiting timbre to a predefined, hardware-specific set of digital instruments.

Theoretical Performance What notation software is most often criticised for is the way in which sonic feedback often comes to be equiparated to (human) music performance. When Leeland Smith presented SCORE as “not a ‘performer’s’ instrument, but rather a ‘musician’s’ instrument,” for example, he claimed that “theoretically, any performance, clearly conceived in the mind, can be realized on [the computer]” (L. Smith, 1972, p. 14). It is indeed a fact that computers can offer automated tasks to an unimaginable extent. However, to translate this type of automation into music composition and performance, results in a disembodied music conception. In other words, an algorithmically generated stream of notes may result in physically impossible tasks for a performer, or for the listener. This is the point of inflexion when envisioning goes beyond the threshold of embodiment. It can be argued, however, that further developments in musical performance techniques can be achieved by pushing the limits of bodily skills. Nonetheless, what I am stressing here is the extent to which music composition can be reconfigured by the possibilities data structures have brought to the field. Furthermore, what is at stake with notation-based music software is yet another musical concern that governed most of music software development during the 1980s: style.

0.3.3 Enter Objects



MILLER PUCKETTE, BARRY VERCOE, JOHN STAUTNER, MIT

"A Real-time Music 11 Emulator"

The MIT Experimental Music Studio is near completion on two real-time synthesis programs for an Analogic array processor. Together they emulate much of Music 11. One plays scores polyphonically on a flexible FM instrument, and the other plays Music 11 instruments written by the user monophonically from a clavier keyboard.

Figure 5: A real-time version of MUSIC-11.
A bodiless abstract published at the ICMC (1981) stating that a real-time version of MUSIC-11 was “near completion” by a group at MIT (M. Puckette, Vercoe, & Stautner, 1981).

Max Faster, cheaper, and portable microcomputers with real-time capabilities for audio processing began to appear onstage within institutions such as MIT and Institut de Recherche et Coordination Acoustique/Musique (IRCAM), and a growing interest among composers and programmers circled around real-time computer music software (See Figure 6). Towards the end of the 1980s, after the proliferation of MIDI (Loy, 1985), composers were already incorporating real-time techniques within musical instruments and software (Vercoe, 1984; M. Puckette, 1991). This is the context for Miller Puckette's development of Max for the 4X real-time audio processor at IRCAM (M. Puckette, 1986). With an emphasis on time and scheduling, Puckette devised a new approach towards complexity in computer music software:

> . . . complexity must never appear in the dealings between objects, only within them. Three other features currently in vogue seem unnecessary. First, there is no point in having a built-in notion of hierarchy; it is usually a hindrance. Second, I would drop the idea of continuously-running processes; they create overhead and anything they do can be done better through [input, output] related timing. Third, there should be few defaults. Rather than hide complexity I would keep it visible as an incentive to avoid it altogether. (ibid., p. 43)

Puckette keeps complexity "visible" within the concept of the programming *object*. Furthermore, he removes the notion of hierarchical programming proposing a light-weight, on-the-spot programming practice based on discontinuous processes: "the scheduler keeps the runnable-message pool in the form of a separate queue for each latency" (ibid., p. 46). In other words, the structure of the database was placed *horizontally* along the time axis, and Puckette's efforts were dedicated to optimizing the internal timing of audio processes. Specifically, linked lists are used to keep track of the order of processes that are run, and each process is scheduled according to its own temporality (latency). Thus, the entire network of processes that can be run is maintained in a dynamic list (stack) that can be changed at any time by adding or removing elements (push/pop). The way in which these processes (methods) are called is by messages that can be sent (input/output) by the user or objects themselves.[8] In sum, the object-oriented paradigm was thus applied to the scheduling system, resulting in a ground-breaking implementation that changed the real-time computer music performance scene: "...rather than a programming environment, Max is fundamentally a system for scheduling real-time tasks and managing intercommunication between them" (M. Puckette, 2002a).

Kyma Another powerful example of an object-oriented language for non-real-time music composition is Carla Scaletti' Kyma, developed at the University of Illinois' Computer Based Education Research Laboratory (CERL) (Scaletti, 1987). It is designed as an interactive composition environment for the Platypus digital signal processor. Scaletti's language was hierarchical in its structure, enabling data records to be linked vertically and horizontally. Together, these data structures formed objects, enabling the composer to treat any set of sounds within the composition, and even starting from the composition itself as an object. In such a way: "...the composer could create a 'sound universe,' endow the sound objects in this universe with certain properties and relationships, and explore this universe in a logically consistent way" (ibid., p. 50). Given the "vast amounts of data required for sound synthesis" (ibid., p. 50), Kyma's objective was to fit timbre creation and temporal event lists into the same traversable database underlying the program. Like Puckette and Free, Scaletti's design was aimed at a language that "itself would not impose notational or stylistic preconceptions" (ibid., p. 50).

On one hand, Scaletti based her research on Larry Polansky's Hierarchical Music Specification Language (HMSL), another "non-stylistically based" music composition environment that was "not fundamentally motivated by a desire to imitate

[8] "The scheduler always sends the first message in the lowest-latency non empty queue. When the associated method returns the scheduler sends another message and so on. The only situation in which we need to interrupt a method before it is done is when I/O (including the clock) causes a lower-latency message to appear...In this case the scheduler causes a software interrupt to occur by pushing a new stack frame onto the stack and executing the lower-latency method. When this method returns ... we pop the stack back to the prior frame at latency d_2 and resume the associated method" (M. Puckette, 1986, p. 46).

certain historical compositional procedures" (Rosenboom & Polansky, 1985, p. 224). Polansky's focus was on a language that would "reflect as little as possible musical styles and procedures that have already been implemented —like conventional music notation—" (ibid., p. 224). On the other hand, the "notational bias" (Scaletti, 1987, p. 49) that Scaletti recognized in languages such as MUSIC-V and FORMES (Rodet, Barrière, Cointe, & Potard, 1982; Boynton, Duthen, Potard, & Rodet, 1986), prescribed a very clear division between composition and synthesis which, in turn, was a very difficult and time-consuming "wall" she had to "circumvent" (Scaletti, 1987, p. 49). Therefore, she imagined a language in which the composer could "choose to think in terms of notes and keyboards and staves but in which this structuring would be no easier and no harder to implement than any of countless, as yet uninvented, alternatives" (ibid., p. 49). Both HMSL and Kyma are still in used today, the former with a further Java version by Didkovsky and Burk (Didkovsky & Burk, 2001), and the latter embedded into a commercially available workstation.[9]

Pure Data Ten years after Max, M. S. Puckette (1997) moved on to Pure Data. The commercially available MAX/MSP (Zicarelli, 1998) presents, like Pure Data, the Max programming paradigm (M. Puckette, 2002a). In resonance with the neutrality of the 1980s, Puckette introduced more data structure flexibility as a means to provide a musical instrument without stylistic constraints. Data structures became a more accessible feature for the user to define and edit:

> The design of Max goes to great lengths to avoid imposing a stylistic bias on the musician's output. To return to the piano analogy, although pianos might impose constraints on the composer or pianist, a wide variety of styles can be expressed through it. To the musician, the piano is a vehicle of empowerment, not constraint. (ibid.)

Puckette, therefore, aims to a certain stylistic neutrality, which he represents by the way in which the user opens the program: a blank page: "no staves, time or

[9] https://kyma.symbolicsound.com/

key signatures, not even a notion of 'note,' and certainly none of instrumental 'voice' or 'sequence"' (ibid.). While acknowledging that even the 'blank page' is a culturally loaded symbol referring to the use of paper in Western Art Music (much in the same way that it is favoring complexity altogether), Puckette reconfigured computer music design, composition, and performance by considering the way in which the structure of the program resonates aesthetically.

Graphic Scores In order to include graphic scores for electronic music within the Pure Data, Puckette implemented a data structure deriving from those of the C programming language, which can be used in relation to any type of data: "the underlying idea is to allow the user to display any kind of data he or she wants to, associating it in any way with the display" (M. Puckette, 2002b, p. 184). Puckette's philosophy, as I have mentioned earlier, was aimed at detaching music software from music concepts, leaving these aesthetic decisions to the user. To this end, anything within the canvas can be customizable, and there is no notion of time assigned to canvas coordinates. However, Puckette provided the user with a sorting function, "on the assumption that users might often want to use Pd data collections as x-ordered sequences" (ibid., p. 185). In fact, this is the only sorting function within Pure Data, and it is the same function that sorts the patch 'graph,' only now made accessible to the user. A common and elementary database routine (`sort`) that emerged to the program's surface because of traditional music notation practices.

Puckette contextualized his research with the *Animal* project by Lindemann and de Cecco which allowed users to "graphically draw pictures which define complex data objects" (Lindemann, 1990), three cases of graphic scores used to model electroacoustic music: Stockhausen's *Kontakte* and *Studio II*, Yuasa's *Towards the Midnight Sun*, and Xenakis' *Mycenae Alpha*, and, most interestingly, the SSSP's user-defined features for graphical representations. Although in the Max papers Puckette does not quote Buxton's research, the latter's numerous publications at ICMC towards the end of the 1970s suggests that they reached the scope of MIT's Experimental Studio where Puckette studied with Barry Vercoe. Furthermore, in Puckette's later introduction of graphic scores to Pure data (M. Puckette, 2002b) (See 0.0.8), he references the SSSP quoted by Curtis Roads (1985) as one source of inspiration, indicating that at least in 2002 Puckette was aware of Buxton's research. In any case, both Buxton's and Puckette's approaches can be considered musical resonances that go beyond geographical limits, reaching the level of data structures in computer music software.

An interesting point in common, however, between much of the interactive composition programs that emerged during the 1980s is that stylistic neutrality became a leitmotif. Computer music software designers were interested in providing stylistic freedom by user-definability. This became a programming need that stemmed from earlier computer music software implementations, and their experimentation. This shift in the course of computer music programs can be understood from two perspectives. On the one hand, by experiencing first-hand the extent to which data structures can indeed structure musical output, the composer-programmers of the 1980s took charge on data structure design and devised new approaches to music-making software. On the other hand, the novel flexibility allowed by the object-oriented model within the programming world made its way to the community by the younger generation of composer-programmers. In any case, the database was moving, expanding through computer music networks, institutions, and softwares.

OpenMusic In the same ICMC 1997 where Pure Data was presented, two object-oriented languages appeared: Realtime Cmix (RTcmix) (Garton & Topper, 1997) and OpenMusic (Assayag, Agón, Fineberg, & Hanappe, 1997). While neither real-time nor a synthesis engine, the strength of OpenMusic resides in its ability to provide the composer access to a variety of sound analysis tools for composition (Bresson & Agon, 2004; Bresson & Agon, 2010), as well as the possibility to generate algorithmic streams that output directly into a traditionally notated score. For example, OpenMusic introduced the concept of a *maquette*, which is a graphic canvas upon which a heterogenous set of elements as varied as audio waveforms, scores, or piano-roll type notation can be displayed. The LISt Processor (LISP)-based graphic language developed as a collaboration at IRCAM held music notation as a focal point, distinguishing it from other stylistically neutral software.

Heaps and Nodes Garton and Topper (1997) presented RTcmix, a real-time version of Paul Lansky's Cmix (Lansky, 1990). What they described as innovative in this project was, in a similar way to the data structures for time management that Puckette presented, the scheduling capabilities of the program. In contrast to the Cmix language, which assumes a non-real-time access of objects, "event scheduling is accomplished through a binary tree, priority-queue dynamic heap. . . " (Garton & Topper, 1997). A heap is a tree-based data structure where both keys and parent-child relationships follow a hierarchical logic. Garton and Topper thus introduced hierarchy into the scheduler. What this allowed, in turn, was "scheduling-on-the-fly," that is, "allowing notes to be scheduled at run-time (usually triggered by an external event, such as a MIDI note being depressed)" (ibid.). The real-time problem became once again a scheduling problem of computational tasks, and it was solved differently with yet another element: instruments instantiated "on-the-fly" could also establish their own Transmission Control Protocol / Internet Protocol (TCP/IP) connection sockets in order to allow for networked access to the individual synthesizers (ibid.). That is to say, whenever a new instrument appears, it has the potential to enter into networked communication with earlier and future nodes. This means that synthesizer nodes could enter and leave the scheduler at any time, always in communication with each other. A musical equivalent would be for a violin player to enter in an out of the orchestra at will, while being able to lend the violin to any other player, and also play any other instrument except the conductor. In a similar networked way, SuperCollider (McCartney, 1996; McCartney, 1998) is a high-level language that provides the user with a different paradigm to handle audio processes scheduling. The innovation that this language implemented, however, is the “garbage collection” of each process. McCartney took the hierarchic structure of the object-oriented paradigm and defined ‘nodes’ in a tree-like structure, each with its own capability of nesting groups of other nodes, but most importantly, with its own initiation and expiration times. In other words, in contrast to Pure Data and MAX/MSP’s constantly running audio processes, SuperCollider only consumes CPU resources whenever it needs to.

Both RTcmix and SuperCollider meant a step forward towards networked musical environments that have resulted in recent forms of music making such as laptop orchestras and live coding, along with new music software such as ChucK (G. Wang & Cook, 2003). The literature on computer music software for composition alone would extend beyond the scope of this dissertation. For further reference in other sound synthesis data structures, see: the Diphone synthesis program (Rodet, Depalle, & Poirot, 1988; Caraty, Richard, & Rodet, 1989; Depalle, Rodet, Galas, & Eckel, 1993; Rodet & Lefèvre, 1996; Rodet & Lefèvre, 1997); the Otkinshi system (Osaka, Sakakibara, & Hikichi, 2002). For an overview of existing audio software up to 2004, see Xamat’s PhD Dissertation (Amatriain, 2004, Chapter 2). See also the Integra project (Bullock & Coccioli, 2009; Bullock, Beattie, & Turner, 2011), and Ariza’s work on python’s data structures (Ariza, 2005).

0.4 intersections

The computer music software race that took place at the level of data structures has moved from music to media in an attempt to generalize applicability by maximizing stylistic potentials. To a certain extent, this motion can be understood as an axis between sound and music data structures. On one hand there is music tradition with its notational baggage. On the other, sound synthesis and programming, with its multi-stylistic promise grounded on the more general use of media. In any case, the shape that this motion takes is given by the composer-programmer's needs, ideas, and implementations. The computer music scene today builds on these struggles, and continues to propose novel approaches that reconfigure the practice.

In this section, I provide a glimpse of the many shapes that this reconfiguration has taken. I focus on artistic ventures, program extensions, and innovative research that has appeared under four main aspects of database performance: corpus-based approaches, querying methods, traversing methods, and resource sharing. These examples point only to some moments in which data structure design changed computer music.

0.4.1 Corpus-based Approaches

Modern uses of databases in computer music take the general form of a corpus of sounds from which descriptors are obtained and then used to create sounds. These are known as corpus-based approaches, also known as data-driven approaches. Their difference is a matter of scale. These approaches have emerged in opposition to rule-based ones, highly useful still in many applications. In what follows, I show some implementations of the corpus-based model in sound.

Concatenative Synthesis Diemo Schwarz developed the concept of data-driven concatenative sound synthesis in his PhD thesis at IRCAM (**Schwarz2000**; **icmc/bbp2372.2003.099**; **Sch06:How**). By segmenting a large database of source sounds into units, a selection algorithm is used to find any given target by looking for "units that match best the sound or musical phrase to be synthesised" (**Sch06:How**). In contrast to rule-based approaches in which sound synthesis is arrived at by models of the sound signal, concatenative synthesis is data-driven, or corpus-driven (when referring to larger databases). That is to say, by joining together recorded samples, Scwharz obtained a model for sound synthesis that preserves even the smallest details of the input signal. Schwarz later contextualized 'information space' as a musical instrument in itself (**diemo_schwarz_2009_849679**; **Schwarz:2012**).

Other approaches The variety of applications of corpus-based or data-driven is still a fruitful research area. I present here only some data-driven cases that arrive at other ways to generate sounds than sample concatenation. **icmc/bbp2372.2003.052** used a database of Short Time Fourier Transform (STFT) analysed sounds in an original way. Upon calculating the distances between the results of these analysis he was able to define a database of similarity between his original database which he then re-synthesized. **DBLP:conf/icmc/Collins07** developed an audiovisual concatenative synthesis method where "databases tagged by both audio and visual features, then creating new output streams by feature matching with a given input sequence" (**DBLP:conf/icmc/Collins07**). A recent case in which concatenative synthesis was applied to rhythm can be found in **Nuannicode225in2016**. **icmc/bbp2372.2003.030** was able to implement a model for heterophonic texture by pitch-tracking the highly ornamented music of the Csángó[10] music into a database that enabled him to present a data structure of the ornament. The implementation of analysis and subsequent algorithmic rule extraction can be thought of as a form of analysis-based sound generation: by inputting a sound file, a dataset of rules was obtained to approach a model for the ornament. Therefore, a rule-model was obtained by means of a data-driven approach. This relates to *Orchidée* (**gregoire_carpentier_2006_849343**), a computer-aided orchestration tool based on database input-matching and a series of candidate orchestration targets. The data-driven approach is combined with a highly dense corpus of instrumental techniques, in order to concatenate orchestral targets.

Software Libraries One of the central concepts of the object-oriented programming is extensibility. The list of objects that can be added to the main program tends to grow exponentially as a function of its use. A list covering all extensions would require a

[10] "The Csángó, in some cases a Szekler ethnic group, are found in eastern Transylvania (Kalotaszeg), the Gyimes valley, and Moldavia" (**icmc/bbp2372.2003.030**).

research project of its own. However, I would like to focus on those extensions that enable further and more specific use of databases in the context of music composition. **Stu04:Mat** developed *MATCONCAT*, a concatenative synthesis library for *Matlab*. **Sch06:Rea** designed Real-Time Corpus-Based Concatenative Synthesis (CataRT) as a concatenative synthesis toolkit both as a standalone application and as a MAX/MSP external. Another concatenative synthesis library is Ben Hackbarth's python module AudioGuide. William Brent's research on timbre analysis developed into a timbre description library for Pure Data called `timbreID` (**icmc/bbp2372.2010.044**). Within this library, users are able to analyze sound files using most available timbre descriptors. Since Brent's library enables users not only to analyze sounds and store the resulting descriptors in a database, but also to cluster them within the database, it allows for a variety of applications of which only one of them is concatenative synthesis.

0.4.2 Querying Methods

Query-by-content One of the innovations that brought forth MIR is high-level audio feature analysis. This enabled computers to understand keywords such as 'bright', 'sharp', 'dark', 'metallic', etc., that would describe timbral content of audio files. When applied to database querying, these keywords enable 'query-by-content' searches. Many online databases such as Freesound or Looperman have this type of querying. The Content Based Unified Interfaces and Descriptors for Audio/music Databases available Online (CUIDADO) project at IRCAM consisted of a database system aimed at content based querying of sound files (**DBLP:conf/ismir/VinetHP02**; **DBLP:conf/icmc/VinetHP02**; **DBLP:conf/icmc/Vinet05**). This project enabled Disk Jockeys (DJs) to browse through files, apply beat-synchronized transitions between them, among other automated tasks during performance. CUIDADO later developed into the *Semantic Hi-Fi* project and influenced subsequent software. **icmc/bbp2372.2007.117** enabled users generation of personalized audio description databases that could also be queried by content in *Data*

Jockey.

Similarity-based **Frisson2015** provides an overview of multimedia browsing by similarity. Real-time audio analysis moved users beyond descriptive keyword, with sound based input by singing or by providing a sample array. These systems calculate the spectral similarity between the incoming signal to obtain a match from a sound database. In this sense, a different type of performativity was enabled with systems with query-by-content in live contexts. For example, *SoundSpotter* (**DBLP:conf/icmc/CaseyG07**) was dedicated to real-time matching of audio-visual streams by using audio input as feed for a shingling algorithm based on Log Frequency Cepstral Coefficientss (LFCCs).[11]

Querying a database by similarity appeared in contexts other than performance workstations. Based on both the *Semantic Hi-Fi* and *SoundSpotter* projects, **Price2008** developed an installation with an interface to a relational database of percussive sounds.[12] This database contained description data of the beginning of each analyzed sound file. Thus, participants were able to query a bank of percussion timbres based on brightness, noisiness, and loudness.

Concatenative synthesis uses similarity for the purpose of sample concatenation at the analysis frame level. In this sense, concatenative synthesis can be understood as a real-time query-by-content engine feeding a granular synthesis engine. Therefore, the difference between concatenative synthesis and content-based-queries is a matter of scale (samples as opposed to sound files) and in the use (new sample combinations as opposed to previously stored sound files).

Hybrid Queries Some authors have managed to conjugate disparate database uses by hybridizing the queries. The following modal translations represent only some of the

[11] "Audio Shingling is a technique for similarity matching that concatenates audio feature vectors into a sequence of vectors, and matches the entire sequence" (**DBLP:conf/icmc/CaseyG07**). "Shingles are a popular way to detect duplicate web pages and to look for copies of images. Shingles are one way to determine if a new web page discovered by a web crawl is already in the database" (**DBLP:conf/ismir/CaseyS06**).

[12] In their project, they used a MAX/MSP library called *net.loadbang-SQL* to query and import data for the communication with Structured Query Language (SQL) databases.

many examples in the literature. **icmc/bbp2372.2001.103** used audio analysis to obtain gesture features from the non-audio signals obtained from the *Radio Drum*, an variant of Max Mathews's *Radio Batton* (**DBLP:conf/icmc/Boie89**). **icmc/bbp2372.2001.103** searched for peak detection in the incoming signal to determine mallet (air) strokes. At CCRMA, **icmc/bbp2372.2001.071** managed to invert the concept of physical modeling by estimating violin bow position, pressure, and speed using Linear Predictive Coding (LPC) coefficients of violin audio recordings. **DBLP:conf/icmc/OliverJ08** presented a controller composed of an elastic head suspended along the rim of an empty drum shell. The player presses the head making different shapes with the hand, fingers, or mallets, and "these shapes are captured by a video camera that sends these images to the computer, which analyzes them and outputs the tracked parameters" (**DBLP:conf/icmc/OliverJ08**). This instrument enabled a possibility for "dissociating gesture with sound" (**DBLP:conf/icmc/OliverJ08**). That is to say, gestures whose sound could be visually anticipated (the hitting of a drum) were extended by micro-gestures only visible to the camera sensor. Further, in contrast with acoustic drums, with the silent drum "one can manipulate continuous sounds through a new gestural vocabulary (**DBLP:conf/icmc/OliverJ08**). Oliver La Rosa developed the sensing algorithm into a Graphics Environment for Multimedia (GEM) external called `pix_drum`, and later moved on to `pix_mano`, where he removed the fabric and focused on what he calls "direct hand-tracking" (**DBLP:conf/icmc/OliverJ10**) (See **??**). **Caramiaux2011** proposed gestural input for the query-by-content method. They used gesture-to-sound matching techniques based on the similarities of temporal evolution between the gesture query and the sound target. Another example of hybrid querying is **mcartwright:2014**, where a database of computer synthesis parameters was queried by vocal input, enabling users to mimic sounds with their voices in order to obtain parameter settings (presets) that would approach the analyzed vocal sound.

0.4.3 Traversing Methods

Given that querying methods have resulted in novel ways to approach information space within databases, many authors have proposed their own approaches towards navigating this space. Like browsing, or surfing the Internet, database traversing is a form of navigation across the *n*-dimensional space that databases have to offer. Despite their differences, the approaches I refer to now point to the hybrid qualities that data can take when used in performance, specifically in terms of the mixed use of data coming from multiple sensing mechanism, and the networked quality that reconfigures music performance and composition.

Sensorial Networks Insook **icmc/bbp2372.2000.146** presented an interactive installation (**Cho00:Voi**) at the Dorsky Gallery in NYC where a 'sensorial network' made from a sound database of speeches by famous leaders was distributed along the installation space. **icmc/bbp2372.2000.146** implemented a motion tracking computer vision algorithm enabled sounds to be modulated as a function of the different 'clouds' of pixel data where values gradually changed as participants moved across the sensing area: "pixels do not switch on and off, they fade in and out forming clusters in the 2D camera plane according to the degree of movement projected from the corresponding floor positions" (**icmc/bbp2372.2000.146**). In this sense, participants were able to walk the database itself: "Traversing the [sensorial network] can be thought of as rotating its shadow such that one moves through a semantic neighborhood which includes sound synthesis and residual tuning as well as speech acts" (**icmc/bbp2372.2000.146**) In addition to this tracking system, however, she included hysteresis within the system. Thus, the recorded history of the participant's interaction with the system enabled condition-dependent events to occur as participants' interaction lasted longer. Within this installation, the artist prototyped a "sensory information retrieval system where the acquisition of information is an acquisition of an experience" (**icmc/bbp2372.2000.146**).

Involuntary Navigation Bioinformatic data taken from galvanic skin sensors attached to a cellist's toes within a live performance environment is the point of departure for a complex network for performance (**icmc/bbp2372.2006.123**). The Galvanic Skin Response (GSR) activity was correlated with intervallic distance between adjacent musical notes in a database of 'cell nodes' previously written by the composer. However such score acted as a "filter for the autonomic control signals generated by the performer" (**icmc/bbp2372.2006.123**). What this means is that the music fragment database, involuntarily navigated by the performer, becomes a parameter for a live-generated score. The performer is thus embedded within a convoluted networked loop that goes through voluntary and involuntary agents that intertwine composition, interaction, and performance.

Networked Collaborations Among the many cases of network performances with multiple players that exist in the literature, I would like to point to one case where the rules of 16th century counterpoint demanded a relational database (**Nakamoto2007**). By implementing a database system such as MySQL, to store and retrieve vocal parts, **Nakamoto2007** enabled performers to sing together in canon form from distant locations. Going beyond any notion of anachrony, what is interesting about this approach is the fact that by "using a PC and database server with the internet" two or more performers can engage seamlessly in musical performance (**Nakamoto2007**). Telematic performances have spawned ever since Internet connectivity enabled networked audio and video feeds. Whalley (2014) considers that the listener's body within telematic electroacoustic concerts has been traditionally left out. Therefore, in he devised a set of parameter constraints within these performances, based on musicians who were used as baseline. His argument was grounded on an affective approach towards networked performance, and it is aimed at addressing the limitations that arise from the separation between performer and listener, specifically within telematic electroacoustic performances.

Mobile Devices The mobility that networks enabled can be represented in the work of **Liu:2013**, who created an audiovisual environment for live data exploration that implemented simultaneous sonifications and visualizations of networked database queries made by participants using I Operating System (iOS) devices. **btaylor:2014** implemented centralized database systems to include user-defined interfaces to be saved and shared within their mobile device platform. **Rya17:OnT** presented a work that gives each member of the audience their own instrument through their cell phones. By accessing a website that loads custom synthesizers made with the Web Audio Application Programming Interface (API), the audience becomes the performer in an innovative way. While the title of the work (**Rya17:OnT**) refers to the potentials and the ubiquity of small transducers, the 'score' (source code) of the work lives on a server and travels wirelessly to the audience to become a (mobile) instrument.

These are some of the many examples that point to the many shapes that traversing a database can take. These shapes have given different resonances within the concert and installation spaces, as well as within the performativity of the music involved. Further, the possibilities of these reconfigurations can be seen in terms of a need for sharing resources and experiences through networks.

0.4.4 Resource Sharing

Sharing resources can be interpreted in many ways. On one end, it points to networked environments on which multiple client users connect to a server that provides shared data flow among the network. This is the case of live coding, where multiple users share the same network. Another definition pertains to the data itself, the way that it is formatted, and how to access or edit it: the file format, where users can read the same data. Lastly, the activity of sharing relates to publishing results like in research or academic communities. This is the case of the multiple datasets that exist.[13] In

[13] 'Dataset' differs from 'database' in terms of scale: multiple datasets may reside in a single database.

any case, what is common between these forms of sharing is an entropic and endless plurality.

Multimodal Datasets Among the many datasets that are available (See **??**), a research interest has been growing among gesture datasets. This is the case of a hand drumming gesture dataset that uses data from a two-dimensional pressure sensor that could be compared to the membrane of a drum (**DBLP:conf/icmc/JonesLS07**). **DBLP:conf/icmc/JonesLS07** aimed to provide physical model designers with a collection of six techniques of hand drumming, recorded as matrices at a slow rate (100 Hz) suitable for non-real-time synthesis by way of interpolation into a model for physical modeling of wave propagation called 'waveguide mesh.' Andrew Schmeder (**icmc/bbp2372.2009.005**), stemming from the research at Center for New Music and Audio Technologies (CNMAT) on the OSC format, proposed a real-time application for efficient storage and retrieval of gestural data using the relational model offered by the PostgreSQL Database Managemet System (DBMS). The motivation behind these datasets, besides research is mostly to provide open access to any user with a computer and an Internet connection. **Young2007** created web-accessible databases of gestural and audio data concerning violin bow strokes. **Hochenbaum2010** developed a gestural and audio joint database that enabled identification of a given performer between a group of performers, gaining insight on musical performance itself. These joint databases combining more than one sensing mode are called 'multimodal.' Multimodal databases can be extremely focused —combining different blowing profiles on recorder flutes (along with their sound) (**Garcia2011**)—, or radically plural: listening subjects asked to move as if creating a sound (**fvisi:2017**).

Formats While the purpose of a format is to store as much information as possible, using as little space possible, and in an efficient way so that read and write operations occur seamlessly, formats are the equivalent of database models within files: they can be implemented in endless ways, and they are contingent upon programming decisions (**Ste12:MP3**). One way to categorize formats is based on human readability. Readability of the format is a function of the task at hand and the quantity of the data involved. In cases where the data is very little, for example, a `.pd` file (Pure Data), MetriX in Extensible Markup Language (MetriXML) (Amatriain, 2004), JavaScript Object Notation (JSON), YAML Ain't Markup Language (YAML), or `.bib` (LaTeX bibliography file), data structures can be stored in (text) characters, and thus be readable by humans. In this sense, data does not need to be highly structured. For example, within the *Integra* project, programmers implemented a data format called Videoalive Indexer Extracted Closed Captions and Metadata (IXD), capable of containing sequences, tags and meta-data, and presets, for shared use among different multimedia environments. Their argument for a semi-structured model resided in the semantic richness that can be allocated in opposition to the binary format only readable by machines. To this end, they implemented IXD using the Extensible Markup Language (XML) language (**icmc/bbp2372.2009.012**). In other cases, data is large enough to justify the need for binary format with a simple header such as `.timid` (**`timbreID`**). At this level, by structuring the format and sacrificing human readable semantic richness, faster write and read times are achieved, and less resources are used. However, in the case of larger media files such as audio, image, or video, and also multimodal gesture data, these demand high-performance compression algorithms that reproduce data in 'streams.' Some formats for sound and gesture analysis were standardized in recent years, as is the case of Sound Description Interchange Format (SDIF) and Gesture Description Interchange Format (GDIF), which are widely used in audio analysis software like Sinusoidal Partial Editing Analysis and Resynthesis (SPEAR) and OpenMusic (Bresson & Agon, 2004; **kristian_nymoen_2011_849865**). In one case revealing the extent to which data can reside in multiple combinations, the SDIF format was used for audio spatialization data (Bresson & Agon, 2004). There is still little format standardization within datasets, and in general, the plurality of formats demands database creators to either implement routines to interpret as many formats as possible, or to rely on external libraries for transcoding. In any case, the plurality of formats is almost as great as that of datasets and, to a certain extent, almost as numerous as there are software developers.

Live Coding Live coding has now a long history and it occupies a fair portion of the computer music scene today. In terms of database performance, the practice of live coding in audio or in video exposes both computer technology and art performance in simultaneity to the cutting edges of both worlds. For a more general overview on live coding, see (**nickcollinsphd**; **Col03:Liv**; **Nilson2007**; **Zmo15:Liv**). In this brief section, I would like to point to the work of **croberts:2014**, who implemented within a real-time live coding web-based environment called *Gibber* a centralized database for the storage and quick access of digital instruments that can be prototyped in the environment. This type of on-the-spot database system enables shared access to sound files that have potential use throughout the performance. By means of a networked database, two or more players can grab and record sounds from different locations. Another case of networked situations in live coding is a system that incorporates content based searches (query-by-humming, query-by-tapping) of various Creative Commons (CC) sound databases such as Freesound or to user-defined databases (**nime18-Xambo-b**).

0.4.5 Closing Remarks

The many shapes that database performance has taken over the years can be approached with what I have shown so far. Since many applications of the database in music continues to grow, I have only selected a few areas in which the database has had some agency. One area that I have not included above is that of artificial intelligence for music applications, where databases have been used for training models, and other forms of machine learning. For example, in interactive music systems (**Row92:Int**), in improvisation systems (**DBLP:conf/icmc/AssayagDD99**; **DBLP:conf/icmc/BlochD08**), to model Electronic Dance Music (EDM) patterns (**rvogl:2017**), analog synthesizer parameter settings (**Loviscach2008**). Multimodal datasets have also been used in training (**DBLP:conf/icmc/SchonerCDG98**). Notwithstanding the multiple gaps and omissions that these lines reveal, I believe the plurality of shapes speaks for itself. As I have shown, from bytes to terabytes, from data structures and files to datasets and databases, has had different positions in relation to sound practices. These positions can be summarized in a three-dimensional diagram (See Figure **??**) where the database can be placed along three axes. Visibility of the database can be represented by the sign: positive values indicate visibility and negative indicate invisibility. ‘Negative’ in this context relates only to the sign of the value, and not to any ‘judgment’ whatsoever. If anything, this graph is intended as a metaphor. As I have mentioned earlier, the database is generally the grounds of sonification, so it can be represented by the y-axis. In MIR, the database is next to the databaser, that is, in the x-axis. Finally, I mentioned that the database in computer music is behind the databaser, therefore the z-axis seems appropriate.

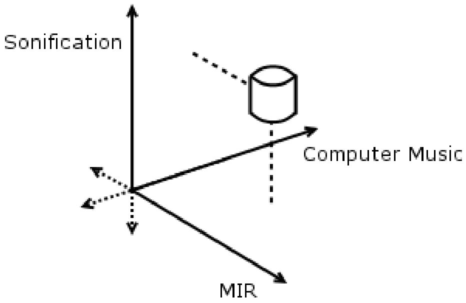


Figure 6: Intersection space.
A snapshot indicating a possible position of the database in terms of visibility among MIR, sonification, and computer music. Positive values indicate visibility and negative indicate invisibility. This graph is only one of multiple frames in which we can capture the visibility of the database in relation to the different practices. Despite the fact that the location in which the database can be placed within this graph is reduced to a single point, the database can occupy multiple points simultaneously. Thus, the more complex coreography inherent in databasing escapes the bi-dimensionality of this graph, but it can nonetheless be hinted by it.

The simplicity of this diagram is intentional, to avoid any attempt to quantize the actual value that the database represents in the plurality of shapes that I have discussed. There is no percentage that can be drawn from how visible a database can be. Therefore, when practices begin to intersect, as I have shown here, the visibility of the database can thus be understood as in constant motion along these axes. Database performance, in this sense, provides a key to understand the motion of this intersection. Furthermore, there is one dimension not contemplated within this diagram: time. The intersections referenced here are always moving in time, which indicates that the diagram that I have shown here is but one frame in the movement of this database choreography. At each point in time the databaser can pause for a second, analyze the frame, and perhaps describe the motion that the database has taken thus far, and the position that it occupies at that point. And further, perhaps the database occupies two positions at once. The dimensionality reduction carried out when graphing the position of the database should not misguide the reader into thinking that the database is fixed, or that it may have only one position. Quite the contrary, the database is in constant motion and incessant fluidity, and databasers engage in a form of dance that sometimes brings the database to a certain visibility, and other times might veil it beyond layers of code. In any case, this description has been my task until now, and it is safe to say that we have glanced at the database dance. In what follows, I will change the focus and approach the database from a different perspective, one not guided by light, but immersed within sound.

0.4.6 Pure Data as Database System

While not technically a database system, Pure Data comprises (internally) a limited amount of data structures that are, nonetheless, different between each other. These structures are, in turn, arrays, linked lists, and symbol tables built as a layer of the C programming language. In terms of database models, Pure Data is mostly hierarchical when it comes to canvases. The windowing system that has a 'root', and multiple 'subcanvases' that can be (almost) infinitely nested. These canvases, while being hierarchic, are traversed as in the navigational model, either for a specific keyword (a query from the 'find' menu), or, most importantly, for signal processing. Besides this hierarchical structure, another important aspect of the GUI level is that it displays visually connected boxes with cords. Therefore, it is quite literally a directed graph where objects are nodes and edges are assigned to a node's inlets and outlets. The `.pd` file format, written in an application-specific language, is structured in such a way that elements on a graph are listed from top to bottom until the end of the list is reached. After this, the connections between objects inlets and outlets are subsequently listed. This graph model, however, comes out of Pure Data's internal design as an object-oriented program. Its core functionality depends on class instantiation. Every internal and external is a class made of C data structures with its own methods, that can be loaded in memory at run time and instantiated any time afterwards. Furthermore, Pure Data is already a networked environment, since in order to effectively 'patch' using the graphical interface, a network is established between Pure Data instance and the Tcl/Tk graphical interface. Added to this, the network capacity that Pure Data comes with, that is, the `pdsend` and `pdreceive` objects that support creation of endless TCP/IP connection sockets, literally exploding the concept of a hierarchical patch into the non-hierarchic, networked model.

A common warning that Pure Data developers have to announce is that if you open a listening port and share your port number, anyone can connect to that port, without any restriction whatsoever.[14] This internet connectivity exposes users to one another in very direct ways, allowing system modifications that if used maliciously could potentially have detrimental effects. It can be argued that this loophole is a reflection of the internal openness of the source code itself. This openness enables programmers to create and load externals, but also to change the program itself. While changing something from the source code can be detrimental for the overall program, in being open, Pure Data prevents any definition to reach completion. A small gap, therefore, is left opened exposing users to the source, and to each other in a networked community.

Pure Data is just one example of many open and non-open source computer music softwares that expose such a plethora of database models for the user. Database models are what makes the realm of data structures reach any databaser: what touches any computer user that has ever pressed a key.

[14] Miller Puckette suggested this during an open discussion at **PdCon16**

Bibliography

Amatriain, X. (2004). *An object-oriented metamodel for digital signal processing with a focus on audio and music* (Doctoral dissertation, Universitat Pompeu Fabra). Retrieved from https://xamat.github.io//Thesis/html-thesis/node51.html

Antila, C., & Cumming, J. (2014). The VIS framework: Analyzing counterpoint in large datasets. In H. Wang, Y. Yang, & J. H. Lee (Eds.), *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, taipei, taiwan, october 27-31, 2014* (pp. 71–76). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T014_162_Paper.pdf

Ariza, C. (2005). *An open design for computer-aided algorithmic music composition: Athenacl* (Doctoral dissertation). Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2017-02-23. Retrieved from http://proxy.library.nyu.edu/login?url=https://search-proquest-com.proxy.library.nyu.edu/docview/305469710?accountid=12768

Assayag, G., Agón, C., Fineberg, J., & Hanappe, P. (1997). An object oriented visual environment for musical composition. In *Proceedings of the 1997 international computer music conference, ICMC 1997, thessaloniki, greece, september 25-30, 1997*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1997.097

Ballora, M. (2000). *Data analysis through auditory display: Applications in heart rate variability* (Doctoral dissertation, McGill University). Retrieved from http://www.markballora.com/publications/diss.pdf

Ballora, M., Panulla, B., Gourley, M., & Hall, D. (2010). Sonification of web log data. In *Proceedings of the international computer music conference, ICMC 2010*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2010.117/1

Barrett, N. (2000a). A compositional methodology based on data extracted from natural phenomena. In *Proceedings of the international computer music conference, ICMC 2000*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2000.123/1

Barrett, N. (2000b). Viva la selva. Retrieved from http://www.natashabarrett.org/viva.html

Beilharz, K., & Ferguson, S. (2009). Aesthetic sonification toolkit for real-time interaction with data. (pp. 401–408).

Ben-Tal, O., Berger, J., Cook, B., Daniels, M., & Scavone, G. (2002). Sonart: The sonification application research toolbox. In *Presented at the 8th international conference on auditory display (icad), kyoto, japan, july 2-5, 2002*, Georgia Institute of Technology. Retrieved from http://hdl.handle.net/1853/51376

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). The million song dataset. In A. Klapuri & C. Leider (Eds.), *Proceedings of the 12th international society for music information retrieval conference, ISMIR 2011, miami, florida, usa, october 24-28, 2011* (pp. 591–596). University of Miami. Retrieved from http://ismir2011.ismir.net/papers/OS6-1.pdf

Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., & Bello, J. P. (2014). Medleydb: A multitrack dataset for annotation-intensive MIR research. In H. Wang, Y. Yang, & J. H. Lee (Eds.), *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, taipei, taiwan, october 27-31, 2014* (pp. 155–160). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T028_322_Paper.pdf

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., ... Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In A. de Souza Britto Jr., F. Gouyon, & S. Dixon (Eds.), *Proceedings of the 14th international society for music information retrieval conference, ISMIR 2013, curitiba, brazil, november 4-8, 2013* (pp. 493–498). Retrieved from http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/177%5C_Paper.pdf

Bortz, B., Jaimovich, J., & Knapp, R. (2015). Emotion in motion: A reimagined framework for biomusical/emotional interaction. In E. Berdahl & J. Allison (Eds.), *Proceedings of the international conference on new interfaces for musical expression* (pp. 44–49). Baton Rouge, Louisiana, USA: Louisiana State University. Retrieved from http://www.nime.org/proceedings/2015/nime2015_291.pdf

Boynton, L., Duthen, J., Potard, Y., & Rodet, X. (1986). Adding a graphical user interface to FORMES. In *Proceedings of the 1986 international computer music conference, ICMC 1986, den haag, the netherlands, october 20-24, 1986*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1986.025

Bresson, J., & Agon, C. (2004). Sdif sound description data representation and manipulation in computer assisted composition. In *Proceedings of the international computer music conference, ICMC 2004*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2004.004/1

Bresson, J., & Agon, C. (2010). Processing sound and music description data using openmusic. In *Proceedings of the international computer music conference, ICMC 2010*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2010.129/1

Brinkman, A. R. (1981). Data structures for a music-11 preprocessor. In *Proceedings of the international computer music conference, ICMC 1981*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1981.018/1

Brinkman, A. R. (1982). Original version of the score11 manual. *Score11 Manual*. Copyright <c> 1982 by Alexander R. Brinkman. Retrieved from http://ecmc.rochester.edu/ecmc/docs/score11/index.html#Introduction

Brinkman, A. R. (1983). A design for a single pass scanner for the darms music coding language. In *Proceedings of the international computer music conference, ICMC*

1980, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1983.002/1

Brinkman, A. R. (1984). A data structure for computer analysis of musical scores. In *Proceedings of the international computer music conference, ICMC 1984*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1984.033/1

Brzezinski-Spiczak, M., Dobosz, K., Lis, M., & Pintal, M. (2013). Music files search system. *CoRR*, *abs/1309.4345*. arXiv: 1309.4345. Retrieved from http://arxiv.org/abs/1309.4345

Bullock, J., Beattie, D., & Turner, J. (2011). Integra live : A new graphical user interface for live electronic music. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 387–392). Oslo, Norway. Retrieved from http://www.nime.org/proceedings/2011/nime2011_387.pdf

Bullock, J., & Coccioli, L. (2009). Towards a humane graphical user interface for live electronic music. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 266–267). Pittsburgh, PA, United States. Retrieved from http://www.nime.org/proceedings/2009/nime2009_266.pdf

Buxton, W. (1977). A composer's introduction to computer music. *Interface*, *6*, 57–72.

Buxton, W. (2016a). Objed: The sssp sound editing tool. *Youtube*. Retrieved from https://www.youtube.com/watch?v=pUoHc_2wUjY

Buxton, W. (2016b). Socializing technology for the mobile human. keynote, the next web conference, amsterdam/europe. *Youtube*. Retrieved from https://youtu.be/rEeEofRShAQ

Buxton, W., Fedorkow, G., Baecker, R., Reeves, W. T., Smith, K. C., Ciamaga, G., & Mezei, L. (1978). An overview of the structured sound synthesis project. In *Proceedings of the 1978 international computer music conference, ICMC 1978, evanston, illinois, usa, 1978*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1978.031

Buxton, W., Patel, S., Reeves, W. T., & Baecker, R. (1980). "objed" and the design of timbral resources. In *Proceedings of the 1980 international computer music conference, ICMC 1980, new york city, usa, 1980*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1980.006

Buxton, W., Reeves, W., Baecker, R., & Mezei, L. (1978). The use of hierarchy and instance in a data structure for computer music. In *Proceedings of the international computer music conference, ICMC 1978*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1978.012/1

Cádiz, R. F., de la Cuadra, P., Montoya, A., Marín, V., Andia, M. E., Tejos, C., & Irarrazaval, P. (2015). Sonification of medical images based on statistical descriptors. In *Proceedings of the international computer music conference, ICMC 2015*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2015.072/1

Cámara Halac, F. (2018). *A spectral experience: Self convolution and face tracking*. Accepted at ICMC 2018 and SEAMUS 2019 conferences.

Caraty, M. J., Richard, J. C., & Rodet, X. (1989). "vowel recognition in a data base of continuous speech: Experiments with local and global identification principles". *EUROSPEECH*, 2272.

Carlile, S. (2011). Psychoacoustics. In T. Hermann, A. Hunt, & J. G. Neuhoff (Eds.), *The sonification handbook* (Chap. 3, pp. 41–61). Berlin, Germany: Logos Publishing House. Retrieved from http://sonification.de/handbook/chapters/chapter3/

Cartwright, M., & Pardo, B. (2012). Building a Music Search Database Using Human Computation. doi:10.5281/zenodo.850060

Ciardi, F. C. (2004). Real time sonification of stock market data with smax. In *Proceedings of the international computer music conference, ICMC 2004*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2004.124/1

Clements, P. J. (1980). Musical data structures in a multi-use environment. In *Proceedings of the international computer music conference, ICMC 1980*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1980.020/1

Collins, N. (2015). The ubuweb electronic music corpus: An mir investigation of a historical database. *Organised Sound*, *20*(1), 122–134. doi:10.1017/S1355771814000533

Connes, A. (2012). The music of shapes. Retrieved from http://www.alainconnes.org/en/videos.php

Corona, H., & O'Mahony, M. P. (2015). An Exploration of Mood Classification in the Million Songs Dataset. doi:10.5281/zenodo.851021

Correa, D. C., Saito, J. H., & Costa, L. d. F. (2010). Musical genres: beating to the rhythms of different drums. *New Journal of Physics*, *12*, 053030. doi:10.1088/1367-2630/12/5/053030. arXiv: 0911.3842 [physics.data-an]

Correia, N. N. (2010). AV Clash - Online Tool for Mixing and Visualizing Audio Retrieved From freesound.org Database. doi:10.5281/zenodo.849729

Crestel, L., Esling, P., Heng, L., & McAdams, S. (2017). A database linking piano and orchestral MIDI scores with application to automatic projective orchestration. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.), *Proceedings of the 18th international society for music information retrieval conference, ISMIR 2017, suzhou, china, october 23-27, 2017* (pp. 592–598). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/235_Paper.pdf

Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2017). FMA: A dataset for music analysis. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.), *Proceedings of the 18th international society for music information retrieval conference, ISMIR 2017, suzhou, china, october 23-27, 2017* (pp. 316–323). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/75_Paper.pdf

Dehkordi, M. B., & Banitalebi-Dehkordi, A. (2018). Music genre classification using spectral analysis and sparse representation of the signals. *CoRR*, *abs/1803.04652*. arXiv: 1803.04652. Retrieved from http://arxiv.org/abs/1803.04652

Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J., & Moussallam, M. (2018). Music mood detection based on audio and lyrics with deep neural net. *CoRR*, *abs/1809.07276*. arXiv: 1809.07276. Retrieved from http://arxiv.org/abs/1809.07276

Depalle, P., Rodet, X., Galas, T., & Eckel, G. (1993). Generalized diphone control. In *Opening a new horizon: Proceedings of the 1993 international computer music conference, ICMC 1993, tokio, japan, september 10-15, 1993*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1993.038

Devaney, J., Arthur, C., Condit-Schultz, N., & Nisula, K. (2015). Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis. In M. Müller & F. Wiering (Eds.), *Proceedings of the 16th international society for music information retrieval conference, ISMIR 2015, málaga, spain, october 26-30, 2015* (pp. 728–734). Retrieved from http://ismir2015.uma.es/articles/261_Paper.pdf

Didkovsky, N., & Burk, P. L. (2001). Java music specification language, an introduction and overview. In *Proceedings of the 2001 international computer music conference, ICMC 2001, havana, cuba, september 17-22, 2001*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2001.007

Diener, G. (1985). *Formal languages in music theory* (Master's thesis, McGill University, Faculty of Music).

Diener, G. (1988). Ttrees: An active data structure for computer music. In *Proceedings of the international computer music conference, ICMC 1988*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1988.020/1

Diener, G. (1989). Ttrees: A tool for the compositional environment. *Computer Music Journal*, *13*(2), 77–85. Retrieved from http://www.jstor.org/stable/3680043

Diener, G. R. (1992). A visual programming environment for music notation. In *Proceedings of the 1992 international computer music conference, ICMC 1992, san jose, california, usa, october 14-18, 1992*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1992.030

Dinuzzo, F., Pillonetto, G., & Nicolao, G. D. (2008). Client-server multi-task learning from distributed datasets. *CoRR*, *abs/0812.4235*. arXiv: 0812.4235. Retrieved from http://arxiv.org/abs/0812.4235

Donahue, C., Mao, H. H., & McAuley, J. (2018). The NES music database: A multi-instrumental dataset with expressive performance attributes. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 475–482). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/265_Paper.pdf

Donahue, C., McAuley, J., & Puckette, M. (2018). Adversarial Audio Synthesis. *arXiv e-prints*, arXiv:1802.04208. arXiv: 1802.04208 [cs.SD]

Dunn, J. W. (2000). Beyond VARIATIONS: creating a digital music library. In *ISMIR 2000, 1st international symposium on music information retrieval, plymouth, massachusetts, usa, october 23-25, 2000, proceedings*. Retrieved from http://ismir2000.ismir.net/papers/invites/dunn_invite.pdf

Dydo, J. S. (1987). Data structures in the note processor. In *Proceedings of the international computer music conference, ICMC 1987*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1987.045/1

Emmerson, S. (1986). *The language of electroacoustic music*. doi:10.1007/978-1-349-18492-7

Eremenko, V., Demirel, E., Bozkurt, B., & Serra, X. (2018). Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 483–490). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/206_Paper.pdf

Erickson, R. F. (1975). "the darms project": A status report. *Computers and the Humanities*, *9*(6), 291–298. Retrieved from http://www.jstor.org/stable/30204239

Flusser, V. (2011). *Into the universe of technical images*. University of Minnesota Press.

Fonseca, E., Pons, J., Favory, X., Font, F., Bogdanov, D., Ferraro, A., ... Serra, X. (2017). Freesound datasets: A platform for the creation of open audio datasets. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.), *Proceedings of the 18th international society for music information retrieval conference, ISMIR 2017, suzhou, china, october 23-27, 2017* (pp. 486–493). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/161_Paper.pdf

Fox, M. K., Stewart, J., & Hamilton, R. (2017). Madbpm: A modular multimodal environment for data-driven composition and sonification. In *Proceedings of the international computer music conference, ICMC 2017*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/bbp2372.2017.087

Free, J. (1987). Towards an extensible data structure for the representation of music on computers. In *Proceedings of the international computer music conference, ICMC 1987*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1987.046/1

Free, J., & Vytas, P. (1986). What ever happened to sssp? In *Proceedings of the 1986 international computer music conference, ICMC 1986, den haag, the netherlands, october 20-24, 1986*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1986.004

Free, J., & Vytas, P. (1988). The CAMP music configuration database. In *Proceedings of the 1988 international computer music conference, ICMC 1988, cologne, germany, september 20-25, 1988*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1988.014

Frid, E. (2017). Sonification of women in sound and music computing - the sound of female authorship in icmc, smc and nime proceedings. In *ICMC* (pp. 233–238). Michigan Publishing.

Garton, B., & Topper, D. (1997). Rtcmix - using CMIX in real time. In *Proceedings of the 1997 international computer music conference, ICMC 1997, thessaloniki, greece, september 25-30, 1997*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1997.106

Good, M. (2000). Representing music using XML. In *ISMIR 2000, 1st international symposium on music information retrieval, plymouth, massachusetts, usa, october 23-25, 2000, proceedings*. Retrieved from http://ismir2000.ismir.net/posters/good.pdf

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2002). RWC music database: Popular, classical and jazz music databases. In *ISMIR 2002, 3rd international conference on music information retrieval, paris, france, october 13-17, 2002, proceedings*. Retrieved from http://ismir2002.ismir.net/proceedings/03-SP04-1.pdf

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2003). RWC music database: Music genre database and musical instrument sound database. In *ISMIR 2003, 4th international conference on music information retrieval, baltimore, maryland, usa, october 27-30, 2003, proceedings*. Retrieved from http://ismir2003.ismir.net/papers/Goto1.PDF

Guedes, C., Trochidis, K., & Anantapadmanabhan, A. (2018). Modeling Carnatic Rhythm Generation: a Data Driven Approach Based on Rhythmic Analysis. doi:10.5281/zenodo.1422615

Hamanaka, M., Hirata, K., & Tojo, S. (2014). Musical structural analysis database based on GTTM. In H. Wang, Y. Yang, & J. H. Lee (Eds.), *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, taipei, taiwan, october 27-31, 2014* (pp. 325–330). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T059_257_Paper.pdf

Hashida, M., Matsui, T., & Katayose, H. (2008). A new music database describing deviation information of performance expressions. In J. P. Bello, E. Chew, & D. Turnbull (Eds.), *ISMIR 2008, 9th international conference on music information retrieval, drexel university, philadelphia, pa, usa, september 14-18, 2008* (pp. 489–494). Retrieved from http://ismir2008.ismir.net/papers/ISMIR2008_173.pdf

Hashida, M., Nakamura, E., & Katayose, H. (2017). Constructing PEDB 2nd Edition: A Music Performance Database with Phrase Information. doi:10.5281/zenodo.1401963

Hashida, M., Nakamura, E., & Katayose, H. (2018). CrestMusePEDB 2nd EDITION: MUSIC PERFORMANCE DATABASE WITH PHRASE INFORMATION. doi:10.5281/zenodo.1422503

Hauger, D., Schedl, M., Kosir, A., & Tkalcic, M. (2013). The million musical tweet dataset - what we can learn from microblogs. In A. de Souza Britto Jr., F. Gouyon, & S. Dixon (Eds.), *Proceedings of the 14th international society for music information retrieval conference, ISMIR 2013, curitiba, brazil, november 4-8, 2013* (pp. 189–194). Retrieved from http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/85_Paper.pdf

Haus, G., & Pinto, A. (2005). Mx structural metadata as mir tools. doi:10.5281/zenodo.849297

Hildebrandt, T., Hermann, T., & Rinderle-Ma, S. (2014). A Sonification System for Process Monitoring as Secondary Task. In *Proceedings of the 5th ieee conference on cognitive infocommunication (coginfocom 2014)* (pp. 191–196). Vietri sul Mare, Italy: IEEE.

Homburg, H., Mierswa, I., Möller, B., Morik, K., & Wurst, M. (2005). A benchmark dataset for audio classification and clustering. In *ISMIR 2005, 6th international conference on music information retrieval, london, uk, 11-15 september 2005, proceedings* (pp. 528–531). Retrieved from http://ismir2005.ismir.net/proceedings/2117.pdf

Hu, X., & Yang, Y.-H. (2014). A Study on Cross-cultural and Cross-dataset Generalizability of Music Mood Regression Models. doi:10.5281/zenodo.850795

Humphrey, E., Durand, S., & McFee, B. (2018). Openmic-2018: An open data-set for multiple instrument recognition. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 438–444). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/248_Paper.pdf

Jaimovich, J., & Knapp, R. (2015). Creating biosignal algorithms for musical applications from an extensive physiological database. In E. Berdahl & J. Allison (Eds.), *Proceedings of the international conference on new interfaces for musical ex-*

pression (pp. 1–4). Baton Rouge, Louisiana, USA: Louisiana State University. Retrieved from http://www.nime.org/proceedings/2015/nime2015_163.pdf

Jaimovich, J., Ortiz, M., Coghlan, N., & Knapp, R. B. (2012). The emotion in motion experiment: Using an interactive installation as a means for understanding emotional response to music. In *Proceedings of the international conference on new interfaces for musical expression*, Ann Arbor, Michigan: University of Michigan. Retrieved from http://www.nime.org/proceedings/2012/nime2012_254.pdf

Jr., C. N. S., Koerich, A. L., & Kaestner, C. A. A. (2008). The latin music database. In J. P. Bello, E. Chew, & D. Turnbull (Eds.), *ISMIR 2008, 9th international conference on music information retrieval, drexel university, philadelphia, pa, usa, september 14-18, 2008* (pp. 451–456). Retrieved from http://ismir2008.ismir.net/papers/ISMIR2008_106.pdf

Karaosmanoglu, M. K. (2012). A turkish makam music symbolic database for music information retrieval: Symbtr. In F. Gouyon, P. Herrera, L. G. Martins, & M. Müller (Eds.), *Proceedings of the 13th international society for music information retrieval conference, ISMIR 2012, mosteiro s.bento da vitória, porto, portugal, october 8-12, 2012* (pp. 223–228). FEUP Edições. Retrieved from http://ismir2012.ismir.net/event/papers/223-ismir-2012.pdf

Karydis, I., Nanopoulos, A., Papadopoulos, A., Cambouropoulos, E., & Manolopoulos, Y. (2007). Horizontal and Vertical Integration/Segregation in Auditory Streaming: A Voice Separation Algorithm for Symbolic Musical Data. doi:10.5281/zenodo.849469

Kirlin, P. B. (2014). A data set for computational studies of schenkerian analysis. In H. Wang, Y. Yang, & J. H. Lee (Eds.), *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, taipei, taiwan, october 27-31, 2014* (pp. 213–218). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T039_344_Paper.pdf

Klein, J. (1998). The wolves of bays mountain. Published in 2004. Open Space.

Klein, J. (2017). *On my compositional approach*. Lecture given at New York University's Waverly Project, on February 2nd, 2017.

Knees, P., Faraldo, Á., Herrera, P., Vogl, R., Böck, S., Hörschläger, F., & Goff, M. L. (2015). Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In M. Müller & F. Wiering (Eds.), *Proceedings of the 16th international society for music information retrieval conference, ISMIR 2015, málaga, spain, october 26-30, 2015* (pp. 364–370). Retrieved from http://ismir2015.uma.es/articles/246_Paper.pdf

Lansky, P. (1990). The architecture and musical logic of cmix. In *Proceedings of the 1990 international computer music conference, ICMC 1990, glasgow, scotland, september 10-15, 1990*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1990.023

Laske, O. E. 1.-., & Tabor, J. (1999). *Otto laske : Navigating new musical horizons*. Westport, Conn.: Greenwood Press.

Lindborg, P. (2017). Pacific bell tower, a sculptural sound installation for live sonification of earthquake data. In *Proceedings of the international computer music conference, ICMC 2017*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2017.033/1

Lindemann, E. (1990). ANIMAL-A rapid prototyping environment for computer music systems. In *Proceedings of the 1990 international computer music conference, ICMC 1990, glasgow, scotland, september 10-15, 1990*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1990.068

Lodha, S., Beahan, J., Joseph, A., & Zane-ulman, B. (1998). Muse: A musical data sonification toolkit.

Loy, G. (1985). Musicians make a standard: The midi phenomenon. *Computer Music Journal*, *9*(4), 8–26. Retrieved from http://www.jstor.org/stable/3679619

Maxwell, J. B., & Eigenfeldt, A. (2008). A music database and query system for recombinant composition. In J. P. Bello, E. Chew, & D. Turnbull (Eds.), *ISMIR 2008, 9th international conference on music information retrieval, drexel university, philadelphia, pa, usa, september 14-18, 2008* (pp. 75–80). Retrieved from http://ismir2008.ismir.net/papers/ISMIR2008_158.pdf

McCartney, J. (1996). Supercollider, a new real time synthesis language. In *Proceedings of the 1996 international computer music conference, ICMC 1996, hong kong, august 19-24, 1996*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1996.078

McCartney, J. (1998). Continued evolution of the supercollider real time synthesis environment. In *Proceedings of the 1998 international computer music conference, ICMC 1998, ann arbor, michigan, usa, october 1-6, 1998*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1998.262

McCurdy, I., Heintz, J., Joaquin, J., & Knevel, M. (2015). Methods of writing csound scores. *FLOSS Manuals*. Retrieved from http://write.flossmanuals.net/csound/methods-of-writing-csound-scores/

Melucci, M., & Orio, N. (1999). The use of melodic segmentation for content-based retrieval of musical data. In *Proceedings of the international computer music conference, ICMC 1999*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1999.355/1

Meseguer-Brocal, G., Cohen-Hadria, A., & Peeters, G. (2018). DALI: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 431–437). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/35_Paper.pdf

Miron, M., & Janer, J. (2017). Generating Data to Train Convolutional Neural Networks for Classical Music Source Separation. doi:10.5281/zenodo.1401923

Mital, P. K., & Grierson, M. (2013). Mining unlabeled electronic music databases through 3D interactive visualization of latent component relationships. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 227–232). Daejeon, Republic of Korea: Graduate School of Culture Technology, KAIST. Retrieved from http://nime.org/proceedings/2013/nime2013_132.pdf

Mitra, J., & Saha, D. (2014). An efficient feature selection in classification of audio files. *CoRR*, *abs/1404.1491*. arXiv: 1404.1491. Retrieved from http://arxiv.org/abs/1404.1491

Morawitz, F. (2016). Molecular sonification of nuclear magnetic resonance data as a novel tool for sound creation. In *Proceedings of the international computer music conference, ICMC 2016*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2016.002/1

Nagavi, T. C., & Bhajantri, N. U. (2013). An extensive analysis of query by singing/humming system through query proportion. *CoRR*, *abs/1301.1894*. arXiv: 1301.1894. Retrieved from http://arxiv.org/abs/1301.1894

Nagavi, T. C., & Bhajantri, N. U. (2014). Progressive filtering using multiresolution histograms for query by humming system. *CoRR*, *abs/1401.2516*. arXiv: 1401.2516. Retrieved from http://arxiv.org/abs/1401.2516

Nardelli, M. B. (2015). Materialssoundmusic: A computer-aided data-driven composition environment for the sonification and dramatization of scientific data streams. In *Proceedings of the international computer music conference, ICMC 2015*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2015.072/1

Nort, D. V., Jarvis, I., & Palumbo, M. (2016). Towards a mappable database of emergent gestural meaning. In *Proceedings of the international conference on new interfaces for musical expression* (Vol. 16, pp. 46–50). 2220-4806. Brisbane, Australia: Queensland Conservatorium Griffith University. Retrieved from http://www.nime.org/proceedings/2016/nime2016_paper0010.pdf

Osaka, N., Sakakibara, K.-I., & Hikichi, T. (2002). The sound synthesis system "otkinshi": Its data structure and graphical user interface. In *Proceedings of the international computer music conference, ICMC 2002*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2002.039/1

Pauletto, S., & Hunt, A. (2004). A toolkit for interactive sonification. In *Proceedings of icad 04. tenth meeting of the international conference on auditory display, sydney, australia, july 6-9, 2004. ed. barrass, s. and vickers, p. international community for auditory display, 2004.* Retrieved from http://hdl.handle.net/1853/50809

Peron, T., Rodrigues, F. A., & Costa, L. d. F. (2018). Pattern Recognition Approach to Violin Shapes of MIMO database. *arXiv e-prints*, arXiv:1808.02848. arXiv: 1808.02848 [stat.AP]

Pesek, M., Godec, P., Poredos, M., Strle, G., Guna, J., Stojmenova, E., … Marolt, M. (2014). Introducing a dataset of emotional and color responses to music. In H. Wang, Y. Yang, & J. H. Lee (Eds.), *Proceedings of the 15th international society for music information retrieval conference, ISMIR 2014, taipei, taiwan, october 27-31, 2014* (pp. 355–360). Retrieved from http://www.terasoft.com.tw/conf/ismir2014/proceedings/T064_307_Paper.pdf

Poddar, A., Zangerle, E., & Yang, Y.-H. (2018). nowplaying-RS: A New Benchmark Dataset for Building Context-Aware Music Recommender Systems. doi:10.5281/zenodo.1422565

Proutskova, P., Rhodes, C., Wiggins, G. A., & Crawford, T. (2012). Breathy or resonant - A controlled and curated dataset for phonation mode detection in singing. In F. Gouyon, P. Herrera, L. G. Martins, & M. Müller (Eds.), *Proceedings of the 13th international society for music information retrieval conference, ISMIR 2012, mosteiro s.bento da vitória, porto, portugal, october 8-12, 2012* (pp. 589–594).

FEUP Edições. Retrieved from http://ismir2012.ismir.net/event/papers/589-ismir-2012.pdf

Puckette, M. (1986). Interprocess communication and timing in real-time computer music performance. In *Proceedings of the 1986 international computer music conference, ICMC 1986, den haag, the netherlands, october 20-24, 1986*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1986.008

Puckette, M. (1991). Something digital. *Computer Music Journal*, *15*(4), 65–69. Retrieved from http://www.jstor.org/stable/3681075

Puckette, M. (2002a). Max at seventeen. *Computer Music Journal*, *26*(4), 31–43. doi:10.1162/014892602320991356

Puckette, M. (2002b). Using pd as a score language. In *Proceedings of the 2002 international computer music conference, ICMC 2002, gothenburg, sweden, september 16-21, 2002*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2002.038

Puckette, M. S. (1997). Pure data. In *Proceedings of the international computer music conference, ICMC 1997*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1997.060/1

Puckette, M., Vercoe, B., & Stautner, J. P. (1981). A real-time music 11 emulator. In *Proceedings of the 1981 international computer music conference, ICMC 1981, denton, texas, usa, 1981*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1981.036

Queiroz, M., & Yoshimura, G. J. (2018). Relative DTW Embedding for Binary Classification of Audio Data. doi:10.5281/zenodo.1422585

Rodet, X., Barrière, J., Cointe, P., & Potard, Y. (1982). The CHANT project: Modelization and production, an environment for composers including the FORMES language for describing and controlling sound and musical processes. In *Proceedings of the 1982 international computer music conference, ICMC 1982, venice, italy, september 27 - october 1, 1982*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1982.035

Rodet, X., Depalle, P., & Poirot, G. (1988). Diphone sound synthesis based on spectral envelopes and harmonic/noise excitation functions. In *Proceedings of the 1988 international computer music conference, ICMC 1988, cologne, germany, september 20-25, 1988*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1988.033

Rodet, X., & Lefèvre, A. (1996). Macintosh graphical interface and improvements to generalized diphone control and synthesis. In *Proceedings of the 1996 international computer music conference, ICMC 1996, hong kong, august 19-24, 1996*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1996.102

Rodet, X., & Lefèvre, A. (1997). The diphone program: New features, new synthesis methods and experience of musical use. In *Proceedings of the 1997 international computer music conference, ICMC 1997, thessaloniki, greece, september 25-30, 1997*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1997.111

Rosenboom, D., & Polansky, L. (1985). HMSL (hierarchical music specification language): A real-time environment for formal, perceptual and compositional experimentation. In *Proceedings of the 1985 international computer music conference, ICMC 1985, burnaby, bc, canada, august 19-22, 1985*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1985.039

Rossiter, D., & Ng, W.-Y. (1996). A system for the musical investigation and expression of levels of self-similarity in an arbitrary data stream. In *Proceedings of the international computer music conference, ICMC 1996*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1996.085/1

Sanden, C., Befus, C. R., & Zahng, J. (2010). Perception based multi-genre labeling on music data. In *Proceedings of the international computer music conference, ICMC 2010*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2010.003/1

Sapp, C. S. (2005). Online database of scores in the humdrum file format. In *ISMIR 2005, 6th international conference on music information retrieval, london, uk, 11-15 september 2005, proceedings* (pp. 664–665). Retrieved from http://ismir2005.ismir.net/proceedings/3123.pdf

Scaletti, C. A. (1987). Kyma: An object-oriented language for music composition. In *Proceedings of the 1987 international computer music conference, ICMC 1987, champaign/urbana, illinois, usa, august 23-26, 1987*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1987.007

Schlei, K., & Yoshikane, R. (2016). The things of shapes: Waveform generation using 3d vertex data. In *Proceedings of the international computer music conference, ICMC 2016*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2016.056/1

Selfridge-Field, E. (Ed.). (1997a). *Beyond midi: The handbook of musical codes*. Cambridge, MA, USA: MIT Press.

Selfridge-Field, E. (1997b). The score music publishing system. *SCORE*. From Beyond MIDI, The Handbook of Musical Codes. Retrieved from http://scoremus.com/score.html

Serizel, R., Bisot, V., Essid, S., & Richard, G. (2016). Machine listening techniques as a complement to video image analysis in forensics. In *IEEE International Conference on Image Processing* (pp. 948–952). doi:10.1109/ICIP.2016.7532497

Simonelli, L., Delgadino, M., & Halac, F. C. (2017). Hearing the self: A spectral experience. For 2 PS3 Eyecams, multichannel audio, screens and participants. 1440 minutes. Xuhui Art Museum, Shanghai, China: International Computer Music Conference.

Skinner, R. (1990a). Music software. *Notes*, *46*(3), 660–684. Retrieved from http://www.jstor.org/stable/941442

Skinner, R. (1990b). Music software. *Notes*, *47*(1), 91–101. Retrieved from http://www.jstor.org/stable/940555

Smith, J. B. L., Burgoyne, J. A., Fujinaga, I., Roure, D. D., & Downie, J. S. (2011). Design and creation of a large-scale database of structural annotations. In A. Klapuri & C. Leider (Eds.), *Proceedings of the 12th international society for music information retrieval conference, ISMIR 2011, miami, florida, usa, oc-*

tober 24-28, 2011 (pp. 555–560). University of Miami. Retrieved from http://ismir2011.ismir.net/papers/PS4-14.pdf

Smith, L. (1972). Score: A musician's approach to computer music. *Journal of the Audio Engineering Society*, *20*(1), 7–14. Retrieved from https://ccrma.stanford.edu/~aj/archives/docs/all/649.pdf

Stoller, D., Ewert, S., & Dixon, S. (2017). Adversarial semi-supervised audio source separation applied to singing voice extraction. *CoRR*, *abs/1711.00048*. arXiv: 1711.00048. Retrieved from http://arxiv.org/abs/1711.00048

Sturm, B. L. (2002). Water music: Sonification of ocean buoy spectral data. In *Proceedings of the international computer music conference, ICMC 2002*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2002.056/1

Thiebaut, J.-B., Bello, J., & Schwarz, D. (2007). How musical are images? from sound representation to image sonification: An eco systemic approach.

Truax, B. D. (1973). The computer composition: Sound synthesis programs pod4, pod5 and pod6. *Sonological Reports*, *2*.

Truax, B. D. (1976). A comunicational approach to computer sound programs. *Journal of Music Theory*, *20*(2), 227–300.

Truax, B. D. (1980). The inverse relation between generality and strength in computer music programs. *Interface*, *9*, 49–57.

Tzanetakis, G. [G.], & Cook, P. [P.]. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, *10*(5), 293–302. doi:10.1109/TSA.2002.800560

Tzanetakis, G. [George], & Cook, P. [Perry]. (2000). Marsyas: A framework for audio analysis. *Organised Sound*, *4*(3), 169–175.

Vercoe, B. (1984). The synthetic performer in the context of live performance.

Vicinanza, D. (2006). A Java Framework for Data Sonification and 3D Graphic Rendering. doi:10.5281/zenodo.849321

Vigliensoni, G., & Fujinaga, I. (2017). The music listening histories dataset. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.), *Proceedings of the 18th international society for music information retrieval conference, ISMIR 2017, suzhou, china, october 23-27, 2017* (pp. 96–102). Retrieved from https://ismir2017.smcnus.org/wp-content/uploads/2017/10/180_Paper.pdf

Vogt, K., Pirro, D., Rumori, M., & Hoeldrich, R. (2012). Sounds of simulations: Data listening space. In *Proceedings of the international computer music conference, ICMC 2012*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2012.096/1

Walker, B. N., & Cothran, J. T. (2003). ICAD 2004: The 13th meeting of the international conference on auditory display, boston, ma, usa, 6-9 july 2003, proceedings. In S. Barrass & P. Vickers (Eds.), International Community for Auditory Display.

Walker, B. N., & Nees, M. A. (2011). Theory of sonification. In T. Hermann, A. Hunt, & J. G. Neuhoff (Eds.), *The sonification handbook* (Chap. 2, pp. 9–39). Berlin, Germany: Logos Publishing House. Retrieved from http://sonification.de/handbook/chapters/chapter2/

Wang, G., & Cook, P. R. (2003). Chuck: A concurrent, on-the-fly, audio programming language. In *Proceedings of the 2003 international computer music conference,* *ICMC 2003, singapore, september 29 - october 4, 2003*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2003.055

Wang, X., & Haque, S. A. (2017). Classical music clustering based on acoustic features. *CoRR*, *abs/1706.08928*. arXiv: 1706.08928. Retrieved from http://arxiv.org/abs/1706.08928

Whalley, I. (2014). Broadening telematic electroacoustic music by affective rendering and embodied real-time data sonification. In *Proceedings of the international computer music conference, ICMC 2014*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2014.046/1

Wilkins, J., Seetharaman, P., Wahl, A., & Pardo, B. (2018). Vocalset: A singing voice dataset. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 468–474). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/114_Paper.pdf

Wilson, C. M., & Lodha, S. K. (1996). Listen: A data sonification toolkit. In *Proceedings of the 3rd international conference on auditory display (icad1996), palo alto, california, november 4-6, 1996. eds.: S. frysinger; g. kramer. international community for auditory display, 1996.* Georgia Institute of Technology. Retrieved from http://hdl.handle.net/1853/50809

Worrall, D., Bylstra, M., Barrass, S., & Dean, R. (2007). ICAD 2004: The 13th meeting of the international conference on auditory display, montreal, canada, june 26-29 2007, proceedings. In S. Barrass & P. Vickers (Eds.), International Community for Auditory Display.

Wüst, O., & Celma, Ò. (2004). An MPEG-7 database system and application for content-based management and retrieval of music. In *ISMIR 2004, 5th international conference on music information retrieval, barcelona, spain, october 10-14, 2004, proceedings*. Retrieved from http://ismir2004.ismir.net/proceedings/p010-page-48-paper227.pdf

Xambo, A., Roma, G., Herrera, P., & Laney, R. (2012). Factors in Human Recognition of Timbre Lexicons Generated by Data Clustering. doi:10.5281/zenodo.850102

Xi, Q., Bittner, R. M., Pauwels, J., Ye, X., & Bello, J. P. (2018). Guitarset: A dataset for guitar transcription. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th international society for music information retrieval conference, ISMIR 2018, paris, france, september 23-27, 2018* (pp. 453–460). Retrieved from http://ismir2018.ircam.fr/doc/pdfs/188_Paper.pdf

Xu, Y., Zang, C., & Yang, J. (2005). Semi-supervised classification of musical genre using multi-view features. In *Proceedings of the 2005 international computer music conference, ICMC 2005, barcelona, spain, september 4-10, 2005*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2005.020

Yang, C. (2001). *Music database retrieval based on spectral similarity* (Technical Report No. 2001-14). Stanford InfoLab. Stanford. Retrieved from http://ilpubs.stanford.edu:8090/489/

Yeh, C., Bogaards, N., & Röbel, A. (2007). Synthesized polyphonic music database with verifiable ground truth for multiple F0 estimation. In S. Dixon, D. Bainbridge, & R. Typke (Eds.), *Proceedings of the 8th international conference on music information retrieval, ISMIR 2007, vienna, austria, september 23-27, 2007* (pp. 393–398). Austrian Computer Society. Retrieved from http://ismir2007.ismir.net/proceedings/ISMIR2007_p393_yeh.pdf

Yeo, W. S., & Berger, J. (2005). Application of image sonification methods to music. In *Proceedings of the 2005 international computer music conference, ICMC 2005, barcelona, spain, september 4-10, 2005*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2005.036

Yeo, W., Berger, S., Lee, J., & Zune. (2004). Sonart: A framework for data sonification, visualization and networked multimedia applications. In *Proceedings of the international computer music conference, ICMC 2004*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2004.128/1

Yolk, A., Wiering, F., & van Kranenburg, P. (2011). Unfolding the potential of computational musicology. In *Problems and possibilities of computational humanities - 13th IFIP iwra 2011 ifip wgs.l — international conference on informatics and semiotics in organisations, ICISO 2011, netherlands, july 4-6, 2011. proceedings* (pp. 137–144).

Zicarelli, D. (1998). An extensible real-time signal processing environment for max. In *Proceedings of the 1998 international computer music conference, ICMC 1998, ann arbor, michigan, usa, october 1-6, 1998*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1998.274