# Database Music

## A History, Technology, and Aesthetics of the Database in Music Composition

by

Federico Nicolás Cámara Halac

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

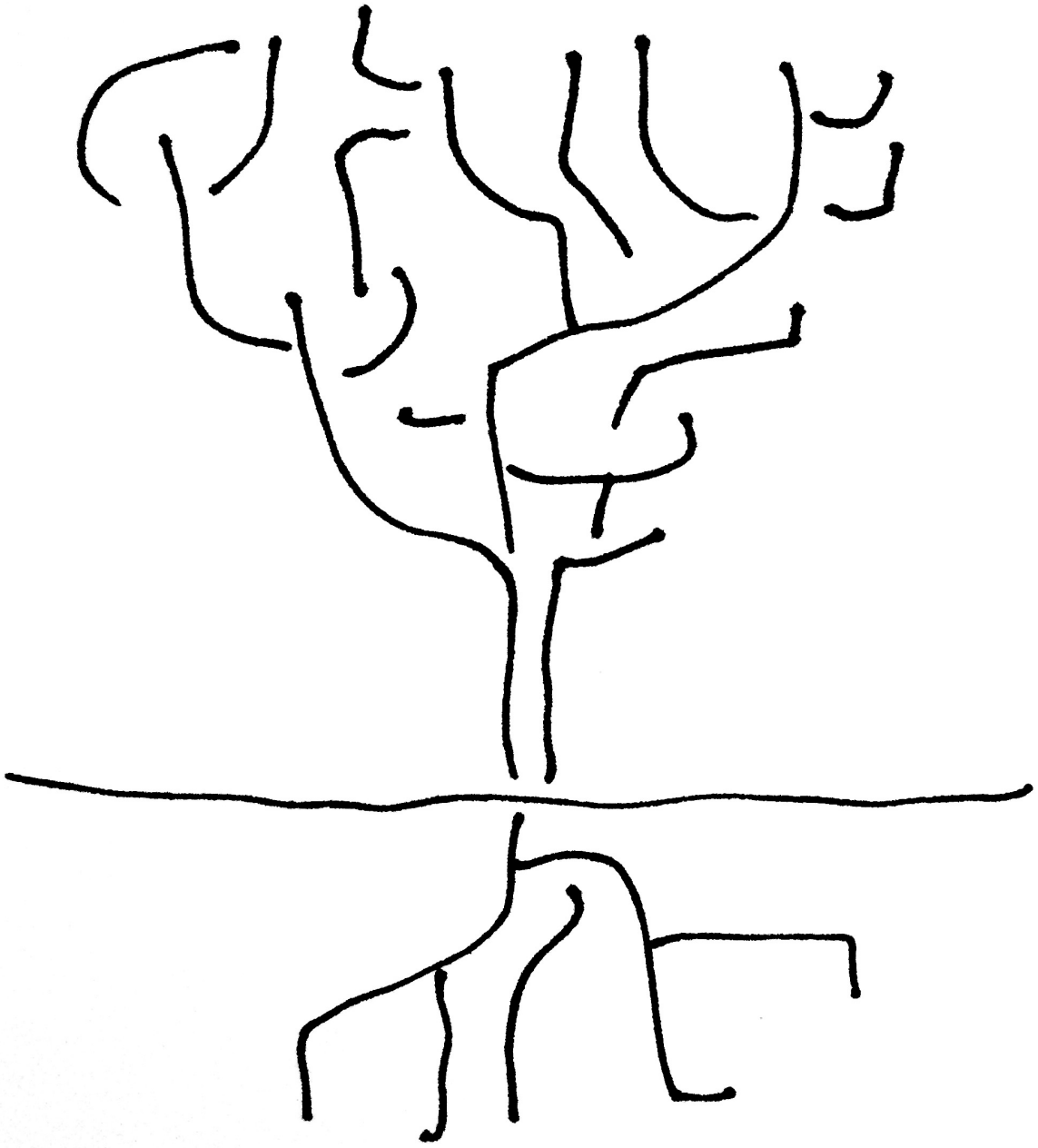
Department of Music

New York University

May, 2019

---

Jaime Oliver La Rosa



# Dedication

For my mother and father, who have always taught me to never give up with my research, even during the most difficult times. Also to my advisor, Jaime Oliver La Rosa, without his help and continuous guidance, this would have never been possible. For Elizabeth Hoffman and Judy Klein, who always believed in me, and whose words and music I bring everywhere. Finally to Aye, whose love I cannot even begin to describe.

# Acknowledgements

I would like to thank my advisor, Jaime Oliver La Rosa, for his role in inspiring this project, as well as his commitment to research, clarity, and academic rigor. I am also indebted to committee members Martin Daughtry and Elizabeth Hoffman, for their ongoing guidance and support even at the very early stages of this project, and William Brent and Robert Rowe, whose insightful, thought-provoking input made this dissertation come to fruition. I am also everlastingly grateful to Judy Klein, for always being available to listen and share her listening. As well as to Aye, for her endless support and her helping me maintain hope in developing this project. I would also like to thank my parents, Ana and Hector, who inspired and nurtured my interest in music from a young age, and my sister Flor and my brother Joaquin who were always with me, next to every word. Finally, this dissertation could not have been possible without the support and help of my friends, some of which I would like to mention by name because they affected directly certain aspects of this text. I would like to thank Matias Delgadino and Lucia Simonelli, for their continuous layers of abstraction; Matias Borg Oviedo, for those endless conversations; Ioannis Angelakis, for his glass sculptures.

# Preface

**Dataloquy (you don't need to read this)** (The initial title that explains how databases are everywhere) The name of the database. (Now think of the data, and the base, and how these relate) These words must point to two things placed one inside the other. (Is it base in data or is it data in base?) The base (of the data). A basement, a basis, a basic foundation for data. The house where data resides. (Is it the base or the data that is economical? Or both?) The addresses in which they are located. (Clearly, you are talking about pointers) The discretized space that guards data. (Guardians of space? This starts to look like a bad sci-fi thing...) Data as in the plurality of datum, and database as the plain (*planicie* in spanish) or the lattice upon which the address space is laid for data. (Data under house arrest) Datum as in bit, as the zero or the one, and nothing in between (Are you sure there is nothing in the middle?) Data as in bytes, and the eight bits that follow it around (like ducklings without mom [*pata* in spanish]) Data as in data types, the many names of the binary words representing the values of almost all numbers (and this 'almost' is still more than enough (Some would disagree)). Data as in data structures and their unions, symbols, lists, tables, arrays, sequences, dictionaries, simultaneously pointing to their interfaces and their implementations and assemblage (The assembly is in order) Data as in files *fichiers* in french, *archivos* in spanish, and their kilobytes and megabytes (inside directories and folders, etc.) Data as in data flow and data streams (are there data fountains?) Data you translate from slot to slot, transmit from client to server to client, transduce to and fro with `adc` and `dac`, transcode from format to format (transgress from torrent to torrent) Data as in dataset for your algorithms to test,

to improve, to fit, to make them more efficient, to teach them the right tendencies, to drive your models data-driven (You are driving me crazy) Data as in data banks (also its transactions and currency) Data as in data corpus (Oh, so it has body?) Data as in database (finally), basing gigabytes with models meant for system management and wearhouses, repositories, their mining, and their subsequent data clouds, clusters, spacing out into the (in)famous big data leap from bit to big.

# Abstract

The aim of this dissertation is to understand the aesthetic agency of the database in music composition. I place my dissertation in relation to existing scholarship, artists, and developers working in the fields of music composition, computer science, affect, and ontology, with emphasis on the ubiquity of databases and on the need to reflect on their practice, particularly in relation to databasing and music composition. There is a database everywhere, anytime, always already affecting our lives; it is an agent in our aesthetic and political lives just as much as we are agents in its composition and performance. Database music lives in between computers and sound. My argument is that in order to conceptualize the agency of the database in music composition, we need to trace the history of the practice, in both its technical and its artistic use, so as to find nodes of action that have an effect on the resulting aesthetics. Therefore, this dissertation is composed of two main sections.

In the first section, I trace a history of database practices from three points of view. The first is from new media theory, emphasizing certain aspects of embodied theory which relate to the intersection between the database and the body. The second is from the history of the database in computer science, giving a panoramic view of the tools and concepts behind database systems, models, structures. The third is from their use in sound practices, describing different approaches to databasing from the fields of music information retrieval, sonification and computer music.

In the second section, I discuss this agency under the broader concepts of sound, self, and community. These three axes are addressed in four sections, each with a different perspective.

First I focus on listening, delineating Jean-Luc Nancy's ontology of sound in order to present the database as a resonant subject in a networked relation and community with the human. Second, I focus on memory, comparing human memory and writing with digital information storing, thus relating databasing and composition with memory, archives and their spectrality. Third, I analyze the performativity of databasing, understanding the database as gendered, in its temporality, repetition, and in its contingent appearance as style, skin, and timbre. Finally, I revise the notion of music work, reflecting on the consequences of the anarchic and the inoperative in the community of database music.

# Contents

# List of Figures

# List of Tables

# Introduction

This dissertation begins with a word ('database') and a proposition: Is there something we can call database music? This sudden jump from noun to adjective comes not without its audible retaliation, and I make no attempt to muffle it. The reader would perhaps forgive the clumsiness of my condition of composer in the midst of writting his dissertation, which made me jump to quickly at the opportunity to make some sound in this initial gesture. Nevertheless, there is a sound and we can listen to it.

In the literature that I have mined (for this dissertation is not only a text, but the sedimented layers of text that I initially traversed with keywords in database queries, such as "`database AND music`"), the database has many histories and many names attached to it. I make no attempt to cover all of these, but I admit that I have (foolishly) tried: the subtitle does begin "a history. . . ." However, it is the 'a' that would account for the gaps and missing parts to which this text is inevitably bound. It is also the 'a' that accounts for the path that I begin delineating accross the two nodes that compose the focus of this text: *database and music*. Both nodes have their historical, technical, and aesthetical idiosyncracies. The primary goal of this dissertation (if I dare to say that it has one) is thus to find where these intersect. The `AND` of the query had indeed much less results than the `OR`, which meant my quest was already promising some reduction. For instance, I indeed decided that the search would only pertain to situations in which computers where involved. Needless to say, not only was my search dramatically expanded through the plethora of database applications, the history of their systematization, and the ongoing struggle between models, it also

opened up the programming world, and with it the history of programming languages, and of the computer itself.

Upon this abysmal enterprise, I did as any musician would and started listening for the sound of databases. I realized that this is not just the sound of your computer reading its hard-drive: it is the sound made with its software. At this point, a network (and this is one of the key terms throughout this text) of sonic software had begun to appear. What this network pointed to, besides the key to open door number one, is a certain silence that much of the literature relating to databases in art continues to abide to: a sonic silence. This relates to the "history, technology…" part of the subtitle. In addressing this silence, I do not attempt to invalidate previous approaches (Manovich, 2001; Vesna, 2007a). On the contrary, these texts have shed light of the key concepts with which I have traversed the sonic software I present, the types of programming decisions I discuss, the various disciplines I place at the intersection of computers and music, and the plurality of shapes that have appeared in relation to databases: door number two. That is to say, through these authors, I introduce notions of embodiment, virtuality, and framing coming from posthumanism (Hayles, 1999) and new media theory (Hansen, 2004), in order to contextualize the role of the database within the practices of MIR, Sonification, and Computer Music. Each one of these disciplines has its own history and it is evidenced by the many conference proceedings and journals that I have (again) mined, as well as the various authors (most of them composers, and most of them programmers) that I refer to. In between these two, that is, in between my exploration of the database in new media theroy, and the its corresponding exploration witihin sound practices, I introduce the more technical evolution of the database, in order to develop a secondary concept that speaks of the performativity of databases: *databasing*. I use this non-existing gerund to refer to all the actions that need to take place around databases, whether these are made by humans or not, refering mostly to these as *databasers*.

After having reached this point, in which the intersection of the database and music was covered in terms of its facticity, as the evidence of a motion, the trace of the database, I could not

help but noticing door number three. I began listening to a certain sound of networks, a certain resonance. This door refers to the next part of the subtitle "...aesthetics of the database..." and opened up to the complex world of sound, flipping this text upside down: *music and database*. (Bang!) The meaninglessness of this reversal, at least semantically or even in terms of a database query, became precisely the point. The sound of databases that I delineate through the first two doors now reached a point where it faces a difference. I enter first with Nancy's (2007) ontology of sound, with which I understand the networks of music software as resonant networks. With this move, the database sounds as a actor (Latour, 1990) that reconfigures the way we think communities (Nancy, 1991). The implicatoins of this association led me to distinguish databases, memory, and archives, and find within their connections a spectral form of authority (Derrida, 1978; Derrida & Prenowitz, 1995). The final move extends towards the activity of the (human) body, and the relationship between database, gender, and performance (Butler, 1988). These three nodes (resonance, spectrality, and performance) encompass an aesthetics of the database that I situate around sound.

Up to this point, my argument might seem to have arrived to an end: I contextualize and define database practices (chapter 1); I develop a technical overview of the performativity of the database into what I call databasing (chapter 2); I review the existing literature with emphasis on sound practices (chapter 3); I conceptualize sound in terms of resonance, networks, and community (chapter 4); I delineate the differences between database, memory, and archive, in order to present the spectrality of databases (chapter 5); I develop databasing in terms of performance, gender, and style in relation to databases (chapter 6). However, there is yet one more leap that the more adventurous reader might take with the final chapter of this dissertation. In chapter 7, I bring the discussion of database and music and database and music...to the latter part of the subtitle "in music composition." With this chapter (a fourth door) I engage with the work of music composition. That is to say, with the history of the database in mind, I rethink the activity of composing (Vaggione, 2001), the role of the composer (G. E. Lewis, 1999), and the operativity of the music

work (Cascone, 2000). The reader will be warned that this last chapter has no conclusions, let alone answers. Neither has it proper questions. It can be held as an attempt to incite, if anything, a provocation before the question.

I have already warned the reader of the clumsiness of a composer in the midst of writing, but that should not discourage neither academic rigor, nor literary thirst. I have thus included an exhaustive bibliography, two interludes, a transcription, and a poslude (work in progress), together with graphs, tables, and some snippets of pseudocode and sometimes working code. In the hope that you will continue reading these p{age,ixel}s, I will (simply) draft an end to these remarks.

# Chapter 1

# Database Art

## 1.1 The Database In New Media Theory

### 1.1.1 Database As Form

> The world appears to us as an endless and unstructured collection of images, texts, and other data records, it is only appropriate that we will be moved to model it as a database —but it is also appropriate that we would want to develop the poetics, aesthetics, and ethics of this database. (Manovich, 2001, p. 219)

To point to the origin of the database as it is known today is not an easy task. Certainly, databases are closely related to the history of computers, but they also relate to the history of lists. The common link between these two is the fact that they are written —on a memory-card, on a page—, which would take its history to the origins of the written word.... However, there is a point where the history of storage takes an operational turn. At this point, the 'word' becomes a type of data, and data begins to bloom exponentially, impulsing faster and more efficient storage and retrieval technologies. Database systems were modelled hand-in-hand with computer languages and architectures from the late 1950s until the present day, when they continue to be developed for almost all aspects of the business world.

In the artworld of the 1990s, the increasing availability of personal desktop computers — with software suites, programming languages, and compilers— resulted in the emergence of new media art. Lev Manovich (ibid.) was the first media historian to argue that the database became the center of the creative process in the computer age. The database had become the content and the form of the artwork in *The Language of New Media*. Furthermore, Manovich recognized that the artwork itself had become an interface to a database; an interface whose variability allowed the same content to appear in individualized narratives. Thus, he claimed that narrative and meaning in new media art had been reconfigured differently. Narrative became the trajectory through the database (ibid., p. 227), and meaning became tethered to the internal arrangement of data.[1] Therefore, for Manovich, the "ontology of the world as seen by a computer" (ibid., p. 223) was the symbiotic relationship between algorithms and data structures. As a consequence of the use of databases in art, the architecture of the computer was transferred to culture at large (ibid., p. 235). Manovich's 'database as symbolic form' thus became a technologically determined shadow that haunted much of new media.

### 1.1.2 A Semiotic Trap

[1] Graham Weinbren writes that a database "does not present data: it contains data. The data must always be in an arrangement... that gives the data its meaning" (Weinbren, 2007, pp. 67–9).

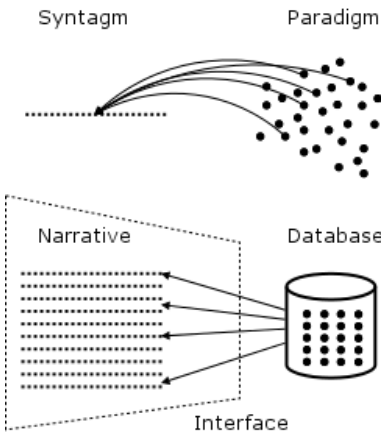



Figure 1.1: Syntagm and Paradigm reversal
Top: syntagm, paradigm, and their relation. Bottom: narrative, database, and their reversed relation.

In order to reveal the extent to which the presence of the database has a radical effect on narrative, however, Manovich reverses the semiotic theory of syntagm and paradigm that governed the first half of the 20th century (ibid., p. 231). Quoting Roland Barthes' reading of Ferdinand de Saussure in *Elements of semiology*, Manovich describes the paradigm as a relation subjected to substitution —because it depends on associations—, and the syntagm as a relation subjected to combination —because it is an instantiation of concrete elements. For example, from the entire set of words in a language (the paradigm) the speaker constructs sentences (the syntagm): the paradigm is implicit (absent) and the syntagm is explicit (present). The relation between these two planes (of the paradigmatic and the syntagmatic) is established by the dependence of the latter on the former: "the two planes are linked in such a way that the syntagm cannot 'progress' except by calling successively on new units taken from the associative plane [i.e., the paradigm]" (Barthes, Lavers, & Smith, 1968, p. 59). Barthes gave several examples with different "systems," one of which was the "food system," which I will borrow in what follows. All the elements that compose a dish, for example, the "set of foodstuffs which have affinities or differences within which one chooses a dish in view of a certain meaning" comes to delimit the paradigm. However, the "sequence of dishes chosen during a meal," or simply, what you are eating as you are eating

it in a restaurant, comes to represent the syntagm (ibid., p. 63). However, when one looks at the restaurant's 'menu', one can glance at both planes simultaneously: "[the menu] actualizes both planes: the horizontal reading of the entrées, for instance, corresponds to the system [i.e., paradigm], the vertical reading of the menu corresponds to the syntagm" (ibid., p. 63). A software menu, for instance, would come to represent both planes as well: the paradigm is the set of all possible actions the user might make within the specific context of the menu; the syntagm is the actual sequence of clicks that the user makes.

Barthes' reading of Saussure is maintained in Manovich's description of the database. On the one hand, narrative is the syntagm since, at least in Manovich's rendition of narrative in the visual art world and the gaming world, it is the trajectory through the navigational space of a database. Furthermore, since this narrative is achieved by the interface, interface and narrative depend on each other: narrative thusly interlocks with the interface itself, and results on the conception of the interface-as-artwork. On the other hand, the database is the paradigm, since it represents the set of elements to be selected by the user. Materially, however, Manovich points to a reversal of these planes. Given the material presence of the database (i.e., the stored data), and given the hyperlinked quality of the user interface, the database becomes explicit (present) and narrative becomes implicit (absent, dematerialised).

For example, consider the case of the typical timeline-view of a video editor.[2] Normally, the user creates a session and *imports* files to working memory, creating a database of files —video files, in this case. Once this database is in working memory, the user places on a timeline the videos, cutting, and processing them at will, until a result is desired, and an *export* or a *render* is made.[3] The timeline where the user places the videos is a visualization of the set of links to the files; an editable graph that allows the user to locate in time the pointers to the elements on the database. This is what Manovich means by "a set of links," because the user is not handling the

[2]This example was used by Manovich in the late 1990s, and it is still valid today with most multimedia editing software.

[3]'Import,' 'export,' and 'render,' refer to processes that read from or write to the computer's disk.

files themselves —as would be the case with an analog video editor, where the user cuts and pastes the magnetic tape—, but the extremely abstract concept of memory pointers.

I consider this reversal to be valid, only on a certain quality of the relation itself, that is, as a shift from one-to-many to many-to-one (See Figure 1.1). The question of the materiality of the database and of the pointers depends on the materiality of data. Links or pointers have, for Manovich, a different (absent-like) status in relation to stored memory itself. This is because of a distinction between pointers and data on the basis of their use: pointers are of a different nature since they do not store data directly. Instead, they refer to the address in memory where a specific stored data begins. However, the mutual binary condition of pointers and data, and the fact that they are both stored in the same memory, reveal Manovich's reversal to be grounded on an equivocation. Pointers are, however functionally different, another data type. This fact comes from the Von Neumann architecture on which computers are constructed (See 1.2.2). If one understands them as moving bodies, it follows that pointers are 'lighter' and travel much faster than other data types, which are 'heavier' and slower to move. However, data types are not moving bodies at all, and thinking of them as such interlocks us in a semiotic trap: accepting this reversal means accepting the materiality of data.

### 1.1.3 Digital Convergence

> A mere 'byproduct' of pleasure, entertainment is a hangover from the media epoch: a function that caters to our (*soon to become obsolescent*) need for imaginary materialization through technology [which, in turn,] serves as a diversion to keep us ignorant of the operative level at which information, and hence reality, is programmed. [emphasis added] (Hansen, 2002, p. 59)

I find in Manovich a silent allegiance to german media theorist Friedrich Kittler's concept of digital convergence. Digital convergence entails that the bodily resonance of media becomes obsolete in the face of absolute digital information storage. Thusly, it turns the human into a "dependent variable" (ibid., p. 59). In the case of physical media, the human body was, for Kittler,

directly shaped by media, and the limit of this 'shaping' was set by the bodily limits of perception. The body became a by-product of media. However, in the age of digital convergence, of an "absolute system of information" (ibid., p. 63), media remove this bodily limit of perception, making the human body a residual product. The body, then, becomes a residue of digital industries.

For example, the extents of this residual aspect of the human can be seen in writer Norman Klein's considerations of the author (N. M. Klein, 2007). Following Manovich's interface-as-artwork, Klein argues that since the reader gets immersed in data, she "evolves pleasantly into the author" (ibid., p. 93). Because the reader participates in the narrative, the result is a reconfigured concept of shared authorship. However, Klein continutes "instead of an ending, the reader imagines herself about to start writing" (ibid., p. 93). This surprising twist in Klein's consideration adds another layer of complexity, namely, the categorical difference between 'writing' and 'not-yet-writing.' In Klein's sense, narrative constitutes a promise of authority that equally blurs the roles of the writer and of the reader. Most importantly, this blurred authority is seen as a reflection of control and subordination of the human. In this view, the potentiality of authority arising from the trajectory through the database belongs neither to the reader nor to the writer: it is appropriated by the database. The roles of the reader and the writer fade into each other and vanish, allowing the database to be a dominant middle term. In other words, human agency is absorbed into a shadow, making the database the sole agent to which the human is subjected. In Klein's own words, the human is a slave to data, and as a consequence the human is economically colonized and psychologically invaded by the evolving force of computers, information, or technology in general (ibid., pp. 86–8). Authority converges, too, in the age of digital convergence.

Media theorist Mark Poster defines technological determinism as the "anxiety at the possibility of [the human mind's] diminution should these external [technological] objects rise up and threaten it" (Poster, 2011, p. X). In other words, the fear or anxiety that the human is ultimately subjected to the power of technology. Understanding new media as digital convergence leads to reading the 'new' in new media as the 'digital.' In reaction to the anxieties that this

convergence brings, and from an embodied approach where databases have an aesthetic agency in resonance with the human, in what follows I propose to shift the focus from narrative (interface) to performance (databasing), and to reconfigure the shadow of the database as a hybrid skin exposing the human and the non.

### 1.1.4 Bodiless Information

> The disembodiment of information was not inevitable, any more than it is inevitable we continue to accept the idea that we are essentially informational patterns. (Hayles, 1999, p. 22)

Media theorist N. Katherine Hayles (ibid.) unearths the theoretical context of cybernetics, upon which the posthuman has been constructed throughout the 20th century. She identifies three waves of cybernetics, each governed by different concepts which helped build the undergirding structures of the technologically determined and disembodied literature in vogue in the 1990s.

The foundational wave cybernetics (from 1945 to 1960) was built, among other concepts, on two main theories: Jon von Neumann's architecture of the digital computer (See 1.2.1) and Claude Shannon's theory of information. As "a probability function with no dimensions, no materiality, and no necessary connection with meaning" (ibid., p. 18), Shannon's formal definition of information within communication systems highlighted pattern over randomness (ibid., p. 33). Therefore, disembodied information became a signal to be encoded, decoded, and isolated from noise.

The word 'cybernetics' [steersman] thus synthesized three central aspects: information, communication, and control. Since the human was seen as an information processing entity, it was "essentially similar to intelligent machines" (ibid., p. 7). Therefore, the conceptualization of the feedback loop as a flow of information came to put at ease notions of human subordination, thus arriving at the governing concept of first wave cybernetics: *homeostasis*. In this sense, the "ability of living organisms to maintain steady states when they are buffeted by fickle environments" (ibid.,

p. 8), became a patch that simultaneously fixed computers as less-than-human, but also pointed to the anxiety of disembodied information that was growing underneath.

However, since the observer of the 'feedback loop' became part of the flow of the system, in the second wave (from 1960 to 1980), cybernetitians reconfigured homeostasis into *reflexivity*, that is, "the movement whereby that which has been used to generate a system [becomes] part of the system it generates" (ibid., p. 8). This became also known as autopoiesis (i.e., self-generation), based on writings by Humberto Maturana and Francisco Varela. This second wave leaves the feedback loop behind, since it considers that "systems are informationally closed" (ibid., p. 10). This means that elements in the system do not see beyond their limits, and the only relation to the 'outside' environment is by the concept of a *trigger*. In this sense, disembodied information was buried deeply into the organization of the system, and the system itself appeared in the form of a cyborg.

Shifting from triggers to artificial intelligence signaled the third wave of cybernetics (from 1980 onwards), whose central concept was *virtuality*. Development of cellular automata, genetic algorithms, and principally, emergence, led to the formation of the posthuman, or an embodied virtuality. However, in Hayles view, the underlying premise of this 'posthuman' is that the human can be articulated by means of intelligent machines (ibid., pp. 17–8). In turn, reconfiguring the concepts of body, consciousness, and technology as inherent to (post-) human life, Hayles argues for the impossibility of artificial intelligence to serve as a proxy for the human. Hayles objective is, then, to dismantle cybernetics from its (relative) assumptions, questioning its major achievements over the years and thereby opening the field for new considerations of the body and its material environment within cybernetics, and by extension, of the body in new media:

> My dream is a version of the posthuman that embraces the possibilities of information technologies without being seduced by fantasies of unlimited power and disembodied immortality, that recognizes and celebrates finitude as a condition of human being, and that understands human life is embedded in a material world of great complexity, one on which we depend for our continued survival. (ibid., p. 5)

While her work is focused on the literary narratives that were built in parallel with cybernetics, she leaves incursions in new media theory for other media theorists. This is where Mark B. N. Hansen comes in.

### 1.1.5 Embodying Databasing

As I describe above, Manovich arrives at this notion of the interface-as-artwork by opposing database and narrative on the semiotic grounds of the reversal of the paradigm and syntagm. In turn, media theorist Mark B. N. Hansen (Hansen, 2004) notes that the interface-as-artwork constitutes a disembodied "image-interface" to information in which the process of information itself (in-formation, giving form) is overlooked. Hansen locates the source of this disembodied conception in Manovich's implicit —but nonetheless evident— premise of the overarching dominance of cinema in contemporary culture, which results in a "disturbing linearity [with] hints of technical determinism" (ibid., p. 36).

For example, Manovich argues that standardization processes originating from the Industrial Revolution have shaped how cinema is produced and received. Attuned to the perceptual limits of the body, the standardization of resolution can be seen (image dimensions, frames per second, and aspect ratio) and heard (audio bit depth, sampling rate, and number of channels). In this sense, the moviegoer and by extension, the listener became industrial by-products, determined by the massively produced electronic devices used for recording and playing. As I have described with Kittler's technological determinism, the devices driven by industrial forces, therefore shaped the body, and as an extension, the aesthetics of cinema.

For Manovich, due to the internal role of the database, the logic of new media is no longer that of the factory but that of the interface. Through the interface to a database, the user is given access to multiplicities of narrative, and thusly, to endless information. The user is granted the power of the database, making in Manovich's eyes the database an icon of postmodern art. In

other words, on an aesthetic level, while mass-standardization and reproducibility of media —the "logic of the factory" (Manovich, 2001, p. 30)— shaped the form of cinema, post-industrial society and its logic of individual customization, shaped the database form. At the bodily level, cinema standardized perception of the passive body, and database individualizes experience. However, this individualized experience still constitutes a technological 'shaping' of the body, a shaping that is exploded into every user quietly sitting behind the screen.

In opposition to this passivity of the body, Hansen describes images as something that emerges out of the complex relationship between the body and some sort of sensory stimulus. In radical disagreement with Manovich, Hansen considers that the image has become a process which gives form to information, and that this process needs to be understood in terms of the body as a filtering and creative agent in its construction. Drawing from Henri Bergson's theory of perception, and in resonance with cognitive science, Hansen defines the function of the body as a filtering apparatus. Under this conception, the body acts on and creates images by subtracting "from the universe of images" (Hansen, 2004, p. 3). Image creation is world creation, and it is not necessarily in contact with the reality that surrounds the body (or the reality of the body), but it is a result of the embodiment of a virtuality that is inherent to our senses. In other words, through this filtering activity, the body is empowered with "strongly creative capacities" (ibid., p. 4). The world is a virtuality that is constructed with our senses and our body. The world can only appear if it appears to the body. Therefore, instead of being a passive node, the body actively *in-forms* data as information (Hansen's word play). The databaser (database user) makes information out of data by precisely embodying the performative act that I call databasing.

### 1.1.6 Filtering And Framing

> The activity in the receiver's internal structure generates symbolic structures that serve to frame stimuli and thus to *in-form* information: this activity converts regularities in the flux of stimuli into *patterns* of information. (Hansen, 2002, p. 76)

The activity of framing, according to Hansen, must be differentiated from that of observation. In this way, "information remains meaningless in the absence of a (human) framer," (ibid., p. 77) and framing becomes a resonance of the (bodily) singularity of the receiver. Quoting MacKay's *Information, Mechanism, Meaning* (1969), the meaning of a message

> . . . can be fully represented only in terms of the full basic-symbol complex defined by all the elementary responses evoked. These may include visceral responses and hormonal secretions and what have you. . . an organism probably includes in its elementary conceptual alphabet (its catalogue of basic symbols) all the elementary internal acts of response to the environment which have acquired a sufficiently high probabilistic status, and not merely those for which verbal projections have been found. (ibid., p. 78)

It is with this conception of framing that Hansen describes precisely that information always requires a frame:

> . . . this framing function is ultimately correlated with the meaning-constituting and actualizing capacity of (human) embodiment. . . the digital image, precisely because it explodes the (cinematic) frame, can be said to expose the dependence of this frame (and all other media-supported or technically embodied frames) on the framing activity of the human organism. (ibid., pp. 89–90)

Therefore, in the context of Kittler's digital convergence, framing prevents the human from being rendered a dependent variable. To the contrary, the framing function of the human body is the possibility condition for the digital to become information. The frame, as Hansen describes, is the human body filtering images from the world, and creating a virtual image that gives form to data. The frame needs to happen as a relation, and thus, it is the temporal instantiation of a process.

### 1.1.7 Database As Aesthetics

The Internet is a place of unlimited access, it is a database in continuous and exponential growth, that reconfigures the grounds on which art has traditionally been built on. One of these grounds is the role of the author. The collaborative approach in the work of media artist Sharon Daniel

(Daniel, 2007), is an example of a different kind of authorial reconfiguration. Daniel raises questions about authority and politics in collaborative art by means of a social model based on the concepts of emergence and *cellular automata*. Cellular automata are systems that reveal emergent (global) behavior from (local) rules. That is to say, each automaton changes states according to its surrounding neighborhood, resulting on an emergent behavior on the global level that precludes top-down behavior. Brought to the social plane, cellular automata result in an inverted hierarchic system: instead of a top-down authority, there exists an emergent, bottom-up behavior. Daniel thus removes her authorial role as artist granting participants a shared authority with the work itself at every stage. Differing from Klein's blurring of the authorial roles, Sharon's participants engage in performative actions that shape the outcome of the artwork in ways she could not anticipate. Therefore, authority is decentered, that is, it does not sediment itself in a single unity.

### 1.1.8 Databasing: Database As Performance

> Data creators have to collect data and organize it, or create it from scratch. Texts need to be written, photographs need to be taken, video and audio need to be recorded. Or they need to be digitized from already existing media... Once digitized, the data has to be cleaned up, organized, and indexed. (Manovich, 2001, p. 224)

Despite Manovich's technologically determined considerations of the database as form, he notes a fundamental aspect of the use of the database when he expresses that data need to be generated (ibid., p. 224). In this sense, he begins to describe the actions that need to be performed around data, or what I call databasing (See 1.2.1), which connotes the use of databases in terms of their performativity. He even goes further and proposes that this activity has become a "new cultural algorithm," (ibid., p. 225) which I reinterpret here as a single subroutine with one argument for input: the world (See Listing 1.1). Following this line of thought, artist Victoria Vesna (Vesna, 2007b) argues that creating a memory bank is a means of testifying to our existence (ibid., p. 25).

```
function new_cultural_algorithm(world)
{
        database ← data ← media ← world
        return database
}
```

Listing 1.1: Manovich's cultural algorithm as a pseudocode routine with the world as argument (i.e, as input). The world is then mediatized, stored in some media (film, tape), then digitized into data, then structured as a database. This routine returns the database, which is, henceforth, the world re-presented as database.

While Manovich calls for an "info-aesthetics" (Manovich, 2001, p. 217), as well as a poetics, and ethics of the database, neither Manovich nor the following generation of media artists and theorists could carry out an exhaustive account of an aesthetics of the database. Several authors continue to abide by Manovich's claim that the aesthetics of the database, or the database as form, is a symptom of the uncritical use of database logic throughout the visual art world of the 1990s. It is in hindsight that his argument can be understood as grounded on the same disembodied constructions that prevent him from including human agency in his account.

## 1.2 Databasing And The History Of Databases

### 1.2.1 Databasing: The Performance Of The Database

> The first step in working with a database is the collection and assembly of the data.... Sorting determines the sequence of presentation, while filtering gives rules for admission into the set presented [,] resulting in a database that is a subset of the "shot material" database. Editing is selecting from the database and sequencing the selections.... To go further: for a filmmaker the term "cutting," as "editing," loses its meaning, and "sorting," "assembling," and "mapping" become more apt metaphors for the activity of composition. (Weinbren, 2007, p. 71)

Like Manovich, Weinbren finds a redefinition in filmmaking impulsed by the selection processes that the database calls for: data collection, generation, and assembly. Weinbren further breaks the selection process into sorting and filtering. With this new terminology, Weinbren makes

a linguistic shift from 'editing' and 'cutting,' to 'sorting,' 'assembling' and 'mapping.' This linguistic shift is significant in the sense that it highlights the practice that is 'under' the filmmaker: databasing.

Databasing is a term I have chosen that best describes the practice of the database, that is, a term that includes the elements and actions of database practices, together with their temporality. The elements of databasing are the different data types and structures that build more complex database systems. The actions of databasing are, on the one hand, the type of operations that a database allows, and on the other, the bodily activity that occur before and after these operations. That is to say, since the operational level occurs below the perceptual threshold of the body, I consider the actions surrounding the immediacy of computations to be defining aspects of databasing.

**Data types and structures**

Depending on the programming language, data types may or may not be part of a data structure, and they store different types of values such as `int, float, char`. These types are then interpreted in binary language by the compiler. Grouping these types into larger sets results in `arrays`. For example, in the C programming language, programmers 'declare' variables first —e.g., `unsigned char age`— and then 'initialize' them with some data —e.g., `age=30`. A simple variable like one's 'age' needs only one value, and given that the `unsigned char` data type only stores values from 0-255, it is safe to use in this case: no age can be negative, no human can live longer than 255 years.

A data structure is a set of data types kept generally in contiguous slots in memory space. It is built for fast allocation and retrieval. A very simple data structure can be thought of as, for example, a person's name together with an age (See Listing 1.2).

```
typedef struct Person {
        unsigned char age;
        char name[128];
} Person;
```

Listing 1.2: An example of a data structure in the programming language C. It is named `Person`, and it holds two variables: `age` and `name`, respectively a positive integer and a string of up to 128 characters.

### Temporality of Databasing

At this point it is important to refer to the higher or lower levels of computer software. A software that is 'higher' means that its simplest operations are composed of multiple smaller operations. The user can thus 'forget' about certain complexities that come from low-level programs, such as memory management. In this sense, low-level programs operate 'closer' to hardware, and programmers need to work at a more granular level. While the above data structure contains low-level features such as setting the size of the name array, it releases the programmer from thinking binary conversion. This means that unless you are changing values directly on the memory card (which is unthinkable), there will most likely be an underpinning software layer.

The speed of regular house computers is so fast that high-level operations happen below the perceptual level (generally below 1-2 milliseconds), hence, for example, the capability for real-time audio processing at high quality sample rates. Therefore, the temporality of activity before and after potentially very large computations feels almost immediate. This means that the body continues almost as if nothing had happened besides a click, or besides the pressing of a key. The immediacy of computation is a feature, certainly, for arriving at extremely fast operations in no time (or zero-time). It is what feels like 'magic' around computers: ask a computer to count to a 1000, and it already has....

However, it may become a bug if we consider the computer as a tool to understand the world. As Manovich claimed, the world understood with computers is not only one that is presented in binary terms, it is one constructed upon a specific set of data structures with their set

of algorithmic rules. The better and more efficient the data structure is, the better and faster the algorithm. In this light, it can be argued that software development is essentially data structure development. At every software release, the software becomes more efficient, using less or more restricted memory space, etc., affecting the scope of its functionality as well as the speed at which it runs. Glancing at the evolution of software in terms of data structure efficiency, therefore, is glancing at a constantly accelerating stream of bits. Because it is immediate, software is incorporated immediately, thus narrowing the temporal window for framing.

This is why the temporality of databasing is context-dependent. As Hansen pointed out, the world can only appear if it appears to the body (See 1.1.5). Data structures, therefore, are very efficient storage devices that have no relation to worlds in themselves, but that are the condition for the possibility of world creating with computers. In this way, the programmer feeds into the computer a notion of world that is then returned by the computer's performance. In each data structure there is a result of a feedback network. One one hand, this network refers to the history of software development, in the sense that each software release is a instance of the much larger event that is software in general. On the other, the network links this history with the practice at hand for which the software is being designed. The sound of a computer music oscillator, for example, even if it were programmed today from scratch, would have embedded histories of computer software design, computer music history, etc.

What is important to note here, is that these interrelations of what is *already there* in software development can be thought of as resonances colliding their way into stability; a stability that emerges not only as a 'stable release' of the code, but also as the condensed multiplicity of worlds that is displaced into a software package. Therefore, data structures are world-making and world-revealing devices that engage with our own capacity for virtuality, and thus they are nodes in our world-making networks.

### Databasing and Writing

As with other new media, the terminology used to describe computer memory is often borrowed from earlier media practices like printed text: reading, writing, and erasing. Computer memory thus shares with writing the property of hypomnesis, that is, of displacing the role of human memory with an external non-human device. In the case of the computer memory however, the scale of this displacement is extremely large, both in terms of the amounts of data that can be stored and the speed with which it can be stored. For example, the 40-bit long 4000 numbers that Von Neumann was aiming at for their memory 'organ' —which was more than plenty for the computational purposes required at the time— represents around 16 Kilobytes, something which today might seem absurd in comparison to current computer storage capabilities that can be found in the case of cloud computing. In light of this fact, we might ask ourselves how is human work transformed through interaction with these massive external memories? Database practice has direct effects on temporality and on memory. Therefore, when designing computer software for art, the way in which data is structured, together with the speed and design of data flow, has significant effects on the temporality of art altogether as a practice.

I have proposed that memory and its storing of instructions and information what enables the computer as such. The simplicity of this synthesis of data and command in Von Neumann's architecture, led to its implementation in not only the computer for which he had intended, also the regular computer as we know it today. Without this architecture, computers would only be able to perform very simple arithmetic operations (like pocket calculators). That is to say, without the computer's ability to store data (the memory organ), the partial differential equations that Von Neumann was aiming at solving would not have been possible. In these equations, the next value of the solution depends on the present value. Therefore, when iterating through every step of the solution, the function in charge of solving the equation needs to access the present value, change it, output the next value, and finally update the present value with the outputted result (See 1.3).

Therefore, in order to provide such solutions, Neumann proposed that: "not only must the memory have sufficient room to store these intermediary data but there must be provision whereby these data can later be removed" (von Neumann & Burks, 1946, p. 3).

```
present ← 0
next ← 0
iteration {
        output ← next ← function() ← present
        present ← next
}
```

Listing 1.3: Pseudocode showing a routine whose next value depends on the present value.

### The Von Neumann Architecture

> Inasmuch as the completed device will be a general-purpose computing machine it should contain certain main organs relating to arithmetic, memory-storage, control and connection with the human operator. It is intended that the machine be fully automatic in character, i.e. independent of the human operator after the computation starts (ibid., p. 1).

Data structures are the turning point of the history of the database. Their appearance enabled the performance of automated algorithms. Within the history of computer technology, data structures begin to appear since Jon Von Neumann's designs of the computer architecture (ibid.). Von Neumann and his team implemented Alan Turing's original concept for a general-purpose computing machine. Of the "certain main organs," it is memory-storage what enables the computer's architecture as we know it today. On one hand, the storage unit of the computer allows data to be written and erased in different locations and times. On the other, the stored data can be not only values to be used during computation, but also includes the algorithmia itself, that is, the commands —functions, operations, routines, etc.— which are used to access and process data for computation. Thus, the interaction of data and command is what defines data flow inside the computer.

Consider, for example, how curator Christiane Paul describes the database as a "computerized record-keeping system", that is, "essentially a structured collection of data that stands in the tradition of "data containers" such as a book, a library, an archive" (Paul, 2007, p. 95). However, when Paul suggests that databases are simply an instance of data collection this only points to the passivity of the container, and not to the potential that it has. An good analogy would thus be a book with the capacity to read itself, if reading were going through every letter in an orderly fashion. A database can also be understood as a library with no need for librarians because all queries are immediate; or, an archive without archeion. These considerations will be developed in the next chapter. While the more general practices of collecting and classifying data are part of the practice of databasing, on some level of the computer architecture, databasing comprises data flow within the Von Neumann architecture. This fact marks a distinction that is better seen in relation to networks. Extending computers via networks like the Internet makes databasing a global activity that expands and changes with every user. This is why I propose that databasing reconfigures the passivity of data containers such as books, libraries, and archives, with a powerful agency that resonates aesthetically.

In order to understand how databases have changed the way we think of earlier types of containers, we need to revise the differences between database models in time. By doing this, I plan to reconfigure the notion of database system. In general, database systems have been used in businesses, namely for administration and transaction. However, narrowing database systems this way raises the similarities or differences between systems to the level of the interface. I propose to delve into the structures of the models to find how the computer itself can be thought of as a database tree, and databasing can be thought of as the activity around databases, or simply: *database performance*. The main purpose of the following account is to understand how computer-based sound practices have participated as a particularly resonant branch of the database tree.

### 1.2.2 A Database Tree

The common use of the word ‘database’ within computer science came around the 1960s, when computers became available to companies throughout the United States of America. For the purpose of data processing, software developers began designing Database Managemet System (DBMS), which are still used in great demand by multiple contemporary companies. The computer’s capability for data processing and storage is inherent in the constitution of database systems. In fields such as Computer Aided Composition (CAC), working with computers meant being part of a system. The human operator has been regarded, for example, as a co-operator (Mathews, 1963). A further approach understands humans operating with computers as another component of complex systems (Vaggione, 2001). In this section, I describe the different levels of database systems as a tree (See Figure 1.2), starting from basic data structures to more elaborate database systems, and then present a brief history of how databases were designed.

jpgjpg

Figure 1.2: A very simple sketch of a tree representing the database tree of computer evolution
0.4

**Ground** The tree is built on different interpretations of the Von Neumann architecture. That is to say, while this architecture went through several optimizations over the years, its three central aspects remained. Therefore, despite the fact that different industry standards for hardware construction resulted in different kinds of operating systems, the core elements of the architecture remained the same: memory (for data and program/code), central processing unit, and input/output interfaces.

**Roots** The below-ground level is accessed through machine and assembly code, which constitutes the core of low-level programming languages and are, to a certain extent, humanly unreadable: the world of bits. Above the ground, readability by humans is the main feature.

**Macros** The database tree metaphor relates to the concept of portability. The database tree only takes the form of a tree once it is instantiated as a software and it is run. That is to say, the database tree unfolds every time it is opened, and in this unfolding it emerges the possibility of dynamically adapting to different grounds. This is what is known in the programming world as defining conditions or macros. With these definitions, their programs can compile with different compilers, across a variety of hardwares and operating systems. Therefore, these database trees have as their main feature the capacity to unfold their roots in different directions upon demand.

**Trunk** The trunk of the tree is composed of data types and structures that provide flow between stored (underground) data and the above-ground components. Programming languages handle data types differently, but in essence, data types and structures are usually built in layers going from the lowest (close to roots) to highest levels.

**Branches** These language layers, after they reach a certain level of complexity, begin to form boughs or limbs that, while being separated from each other, are linked to the same trunk and roots. I consider branches to be programs with text-based interfaces such as Bash, C, C++, python, Java, etc. Their feature is their generic functionality.

**Twigs** More complex programs built on top of branches, such as Pure Data, Supercollider, R, octave, Processing, OpenFrameworks etc., are dedicated for a narrower scope of tasks. Their feature is their level of specialization for the task at hand: sound synthesis, statistics, visuals, etc. They might be more application-specific. In general, these programs are commonly considered layers on top of other languages, libraries, or software frameworks.

**Leaves** User interfaces (or GUIs) are the leaves of the tree. I relate the photosynthetic quality of leaves with user input/output interaction. Despite their simple, user-friendly appearance, software leaves are highly complex systems such as multimedia editors (Adobe Creative Suite or Microsoft

Office), Internet browsers, mobile apps, etc. A particular kind of leave is the DBMS, generally used in businesses for data processing and editing, for example: MySQL, PostgreSQL, Non or Not only SQL (NoSQL), Appache Couch Database (CouchDB) and MongoDB.

**Networks** An important feature of database trees is their network capabilities. Networks can be established by connecting leaves, branches, or roots with each other, both within the same tree and with other trees. For example, software can establish a network between its graphical interface and its core program —as is the case with Pure Data, for example. Another example would be the way in which DBMSs interact with data: the MySQL database model allows the user to load a data set in working memory, and establishes a connection between the opened memory and the input/output mechanisms. Networks of trees are data streams running by way of an Internet Protocol (IP) and a client-server type of relation. Cloud storage services such as Google Drive, ICloud, OneDrive, and Dropbox are used as a networked way to store and share data. One tree can serve as data storage and processing repository, and other client trees can connect to the server tree and request data or processing of data from it. This is the essence of the internet and all the communication services that it enables, such as email services, social networking sites, and multi-user collaboration platforms like Github. This allows software like Pure Data and MySQL to have their respective core program and data sets in one computer, and their interfaces on a different one.

**Clouds** Combining networked databases with computer clusters forms what is known as cloud computing. For example, most universities provide clusters for data processing —e.g., NYU's Prince cluster— that can be accessed from remote locations. These clusters are massive server architectures made out of multiple processing and memory units joined together. These architectures began developing in the 1990s, coining terms like data mining (Kamde & Algur, 2011), data warehouses, data repositories (Silberschatz, Stonebraker, & Ullman, 1995).

### 1.2.3 The Realm Of Data Structures

Data structures are the building blocks upon which the entire database model is designed. A data structure is a way to organize data so that a set of element operations are possible, such as `ADD`, `REMOVE`, `GET`, `SET`, `FIND`, etc. Data structures can be thought of in two ways: either implemented or as interfaces, what is also known as *abstract data types*:

> An interface tells us nothing about how the data structure implements these operations; it only provides a list of supported operations along with specifications about what types of arguments each operation accepts and the value returned by each operation. (Morin, 2019, p. 18)

In other words, the abstract data type represents the idea of the structure. When abstract data types are implemented in code, the speed and efficiency of the data structure can be physically evaluated. An implementation of this sort includes "the internal representation of the data structure as well as the definitions of the algorithms that implement the operations supported by the data structure" (ibid., p. 18). Because of the consequences that design has on computational performance, data structures have constituted a focal research point in the database and computer science communities.

**Array data structure** Arrays constitute one of the oldest and most basic data structures. They are contiguously stored, same-type data elements referenced to by indices. Most programming languages have implemented arrays. Most real-time software loads sound files or images to working memory as an array (or a buffer) of contiguous samples or pixels. Arrays are use less resources when reading than when writing, since accessing their elements is achieved by pointers, but editing demands copying large portions of the array back and forth.

**Linked Lists** One important technical shift in the use of data structures came with the concept of linked lists. A linked list is collection of data (usually a symbol table), with pointers to the 'previous' and/or 'next' item on the list. They are built to maintain an ordered sequence of elements.

This functionality was only available after the FORTRAN '77 programming language (1977) and later it became integrated in the C programming language (Kernighan, 1978). They differ from arrays since they can hold multiple data types (including arrays and other data structures), and they are accessed by traversing the list using the 'previous' and 'next' pointers. In the programs developed during the Structured Sound Synthesis Project (SSSP) and Computer Assisted Music Project (CAMP) years (See 1.3.3), linked lists were used in the (then) very recent C programming language. Ames (1985) as well as Rowe (1992) used linked lists, the former to represent melodies within an automated composition system, the latter within the `Event` data structures of the interactive music system *Cypher*.

**Sequences** Crowley (1998) claims, however, that neither linked lists nor arrays are suitable for large text sequences, since linked lists take up too much memory, and arrays are slow because they requires too much data movement. Nonetheless, he argues, "they provide useful base cases on which to build more complex sequence data structures" (ibid.). In fact, data structures are generally built from arrays and linked lists. For example, in designing *Audacity*, Mazzoni and Dannenberg (2001) implemented the concept of sequences, into a set of small arrays whose pointers were traversed in a linked list. Large audio files were loaded and edited at very fast processing times.

### 1.2.4 A Brief History Of Database Models

I propose now to extend the concept of *abstract data types* to the concept of database *models*. Database models are the realm of data structures. These models, to be described below, constitute the abstract ways in which data can be organized within a database system. DBMSs, in turn, are a specific type of software aimed at organizations, website design, server architectures, company management, among other uses in the business sector. Since an analysis of these systems falls outside the scope of this study, I provide a glimpse of the structure of the models without entering in their implementation. Figure 1.1 shows a development timeline that serves as a context for the

appearance of these models. Their emergence over the years goes hand in hand with hardware and programming language development. Further, several implementations of these models depended on specific language development such as Data Definition Language (DDL) for structural specification of data, and a data manipulation language (DML) for accessing and updating data (Abiteboul, Hull, & Vianu, 1995, p. 4).

Angles and Gutierrez (2008) name the three most important aspects a database model should address: "a set of data structure types, a set of operators or inference rules, and a set of integrity rules" (ibid., p. 2). Operators can be understood as the set of routines that constitute the query language and data manipulation. Integrity rules can be understood as data constraints preventing redundancy or inconsistencies, and checking routines preventing false queries. In a similar way, for Abiteboul et al. (1995) a database model "provides the means for specifying particular data structures, for constraining the data sets associated with these structures, and for manipulating the data" (ibid., p. 28). However, data manipulation (operators) and constraints (integrity) are built around the data structure, which is why, Angles and Gutierrez (2008) continue, "several proposals for [database] models only define the data structures, sometimes omitting operators and/or integrity rules" (ibid., p. 2).

In essence, all DBMSs share the same function: provide access to a database. This access, however, is restricted by the imperatives of the model. Database models have been thought of as collections of conceptual tools to represent real-world entities and their relationships (ibid., p. 1). In this sense, the models are fit to achieve a level of specificity and efficiency that is integrated with the notions of economic success. That is to say, the quality of database access has a direct influence on the operational level of businesses. For example, if the database system in charge of airline reservations fails to update an entry or does not restrict duplicates, this might result in either empty airplanes or double-booking, an economic loss that might result in a company going out of business. In relation to data structure design within Computer-Aided Algorithmic Composition (CAAC) software, Ariza (2005a) claims that design choices "determines the interaction of software

components and the nature of internal system processing" (ibid., p. 18). Luckily, a failed database access in music might perhaps come as a minimal performative 'bump' that can be otherwise forgotten. However, it is imperative that these models are analyzed because of the continuum between data structures and database models, and because of the internal relations that resonate from these structures to the implementations of computer music software. Therefore, to a certain extent, database models and computer music software share the resonance of data structures, and belong to their realm.

**Hierarchical**

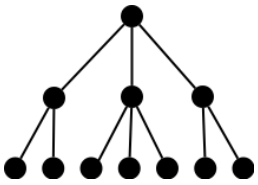


Figure 1.3: Hierarchical Model
Diagram of the hierarchical model

The hierarchical model was developed at International Business Machines Corporation (IBM) during the early 1960s, in conjunction with other American manufacturing conglomerates for National Aeronautics and Space Administration (NASA)'s Project Apollo, resulting in Information Management System (IMS) (Long, Harrington, Hain, & Nicholls, 2000). The hierarchical model is closely linked to the architecture of data within a computer. Therefore, it interprets records as collections of single-value fields that are interconnected by way of paths. Records can have type definitions, which determine the fields it contains. As a rule of this structure, a child record can be linked upwards to only one parent record and downwards to many child records. The structure stems from a single 'root' record, which is the initial parent-less record through which all other records are accessed.

This model is useful for nesting structures such as directory trees and path structures in

most operating systems today. Their use within database systems was eclipsed by the relational model during the 1980s, but it resurfaced through relational-type implementations of hierarchical models, and with the appearance of semi-structured model in the late 1990s (See 1.2.4).

**Network**

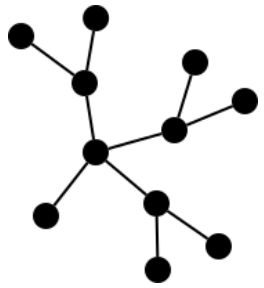


Figure 1.4: Network Model
Diagram of the network model

Invented by Charles Bachman in 1969 and published at the Conference/Committee on Data Systems Languages (CODASYL), the network model is a way of representing objects as nodes in a graph whose relationships can be represented as arcs. The programming language Common Business Oriented Language (COBOL) was designed for the implementation of network databases. The nodes in these networks are known as 'records,' and their relationships form 'sets' that have one-to-many relationships in between records, that is, one 'owner' and multiple 'members.' The main feature of a network model is that these relationships are not bounded to any hierarchical or lattice-like structures, providing a more natural way of record relation. Structurally, each node has an identity called a database 'key' which corresponds to the pointer to the physical address of the record on disk. This is how the network model maintains a close relationship between data structures and their traversal. Traversing the network means going from node to node, that is, keys can be used to implement linked lists for record navigation. These nodes do not have a hiearachical structure, meaning that the network can be accessed starting from any node. Due to the interlocking of the physical implementation and the internal logic of node identity and access,

very fast retrieval speeds are obtained.

**Navigational Paradigm** The advent of disk-based database systems, in contrast to magnetic tape or punched card systems, enabled a different way of thinking database navigation. Working for General Electric's Integrated Data Store (IDS), Bachman (1973) later conceptualized and implemented a navigational paradigm within the networked model. Abandoning the "memory-centered view" of database system development, Bachman called for programmers "to accept the challenge and opportunity of navigation within an *n*-dimensional data space" (ibid., p. 657). Therefore, he proposed data records and attributes as *n*-dimensional space. This means that a database can be traversed not only by accessing the first element and then moving sequentially to the 'next' record. Secondary data keys could be made into sets for navigation starting from any of its members. In other words, given a database with records and attributes, all attributes can become a new dimension thus making retrieval times much more efficient. Navigating through a database within this paradigm is achieved by following record relationships instead of record order in physical storage. Therefore, with the navigational paradigm, a new level of abstraction was thus given to database management systems, resulting in better and more efficient database retrieval.

The navigational paradigm was implemented not only in network model, also in the hierarchical model, and it is still used today. Like I described with hierarchical databases, the navigational paradigm was eclipsed by the relational model, but after the 1990s, they re-emerged with non-relational databases. For example, since Document Object Model (DOM) websites contains a hierarchicalstructure, they can be accessed using this navigational paradigm.

**Relational**

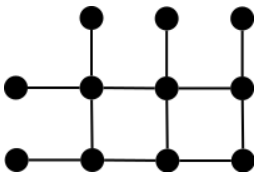


Figure 1.5: Relational Model
Diagram of the relational model

The relational model was first designed by Codd (1970), Codd (1972). Its main feature is the table-like organization of data, together with a separation between the physical level of data storage and the query language. These features allowed, on the one hand simple data visualizations, and on the other highly complex data manipulations by way of an algebra-based query language. Data is placed into uniquely identified rows (records) which can have multiple columns (attributes). A table thus becomes a relation. The main difference between the navigational and the relational paradigms, can be seen in the way users formulate queries. In the former, users specify which steps need to be made in order to arrive at a certain record. In the latter, users specify what needs to be found in terms of an algebraic expression. The query language developed for relational databases is Structured Query Language (SQL). In recent years, object relational database have emerged such as SQLObject, intepreting relations as classes in the object-oriented programming paradigm.

**Non-Relational**

This is a more general type of database models where the internal structure is different from the tabular kind that the relational model presents (See 1.2.4), and they are generally referred to as NoSQL. Within this class or group of non-relational models, some examples can be: Key-Value databases, which are centered on associative arrays (hash tables) such as python dictionaries; semi-structured databases (See 1.2.4), also called document-oriented databases such as Extensible Markup Language (XML), YAML Ain't Markup Language (YAML), and JavaScript Object Notation (JSON); graph databases and mixed graph models such as the way in which the World

Wide Web convention (W3C) structures websites, with a URL as a 'name' and their content as a 'graph' (See 1.2.4); object databases (See 1.2.4); and database systems using combinations of different models.

### Graph

In their survey of graph-modelled databases, Angles and Gutierrez (Angles & Gutierrez, 2008) date the beginning of graph databases to the early 1980s, in conjunction with object-oriented databases. This model interprets records as 'nodes' and connections as 'edges.' Therefore, visualizations as graphs, as well as operations stemming from the mathematical theory of graphs, are features of the model. The visual programming paradigm takes advantage of graph representations of their object-oriented programming structure. In this sense, computer music software like OpenMusic, PWGL, Pure Data, MAX/MSP, Kyma, among others, present their objects as a directed graph on a canvas.

### Object

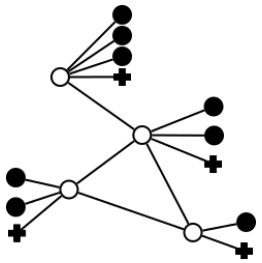


Figure 1.6: Object Model
Diagram of the object model

These databases combine the object-oriented programming paradigm with database concepts. On one side, each record is treated as an object, with capability to store variables (attributes) and functions (methods) that the object can perform. This way, when an object is instantiated in the form of a record, all the attributes and methods become available to itself and to other objects,

provided these are setup in a 'public' way, and so different interactions can occur throughout the database. Some programming languages are directly object-oriented, from which certain databases were created (See 1.1). From 2004, the open source community has been developing open source object databases that are easily accessible in several object-oriented languages.

**Semi-structured**

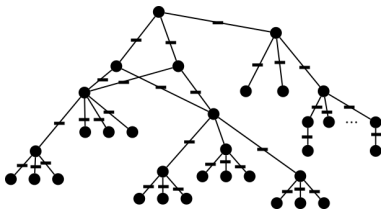


Figure 1.7: Semi-structured Model
Diagram of the semi-structured model

> We call here semi-structured data this data that is (from a particular viewpoint) neither raw data nor strictly typed, i.e., not table-oriented as in a relational model or sorted-graph as in object databases. (Abiteboul, 1996)

Abiteboul (ibid.) comments that given the amount of data that has grown in non-standard structures, a new way of accessing data has emerged. Furthermore, access to data can take place from a variety of different platforms such as browsers, query languages, application-specific interfaces, etc., making the process of obtaining useful information increasingly more difficult since these platforms call for specifically tailored methods and languages. Abiteboul claims, therefore, that first there is a need to extract the non-standard structure from the data, so that it can be traversed afterwards. These databases constitute the semi-structured model. Some examples of this model include XML databases, JSON files, YAML files, among others (Buneman, 1997). A well known database of this kind is the Internet Movie Database (IMDB).

### 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 Graphical User Interface (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 Transmission Control Protocol / Internet Protocol (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.[4] 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. An 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.

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

| Year | Model | Designer | Implementation |
|---|---|---|---|
| 1959 | Hierarchical | IBM | IMS |
| 1960s | Network; Navigational | CODASYL; General Electric; Hewlett Packard (HP); United Information Systems (Unisys) | IDS; Integrated Database Management System (IDMS); Raima Database Manager (RDM); TurboIMAGE; Unisys OS 2200 databases (OS 2200); NoSQL |
| 1960s | Deductive | J. Minker; L. Kuhns | |
| 1960s | Non-relational | Apache; Sparsity; | MongoDB; Redis; Cassandra; Sparksee; NoSQL |
| 1970s | Relational | E.F. Codd; P. Chen (1976) | MySQL; Oracle Corporation (Oracle); PostgreSQL; Microsoft Access (Access); Structured Query Language Lite (SQLite) |
| 1975 | Semantic model | U.S. Air force; J.H. ter Bekke (1991) | XPlain |
| 1980 | Graph | Oracle; Apache; Amazon | Neo4j; Oracle Spatial and Graph; ArangoDB; Amazon Neptune; Boost Software Library (BOOST); NetworkX (NetworkX) |
| 1985 | Object | Brown University; Texas Instruments; Bell Labs; Apache | GemStone (Smalltalk); Gbase (LISt Processor (LISP)); CouchDB; SQLObject |
| 1990s | Semi-Structured | W3C | XML; Sedna |
| 1995 | In-Memory | Oracle; Sybase; Exasol AG; VMWare | TimesTen; Adaptive Server Enterprise (ASE); High Performance Analytics Appliance (SAP HANA); EXASolution; WebDNA |

Table 1.1: Database model development timeline with examples.

## 1.3 Databasing Sound: Applications Of Databases In Sound

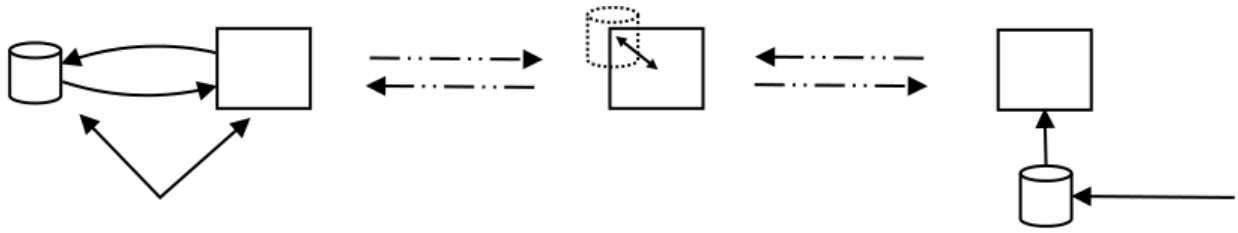


Figure 1.8: Database performance and interdisciplinary feedback.
The arrows between databases (cylinders) and computers (squares) represent data flow. Left: the database is 'visibly next' to the computer, as is the case with MIR; the two bottom arrows indicate the intervention of the human operator. Right: the database is 'visibly below' the computer as is the case with Sonification; the database feeds the computer from an external source (right arrow). Middle: the database is 'invisibly behind' the computer, within the softwares used for (and as) music works. The arrows in between the practices represent interdisciplinary feedback.

Having discussed the current state of new media theory and the theory of databases and data structures, in this section I theorize the use of databases in relation to sound. To a certain extent, ever since the first computers were used to make music the database has been an invisible partner in the music literature. I argue that by sheding some light to this inherent aspect of computers we can arrive at a clearer notion of how databases sound. Praticularly, by placing the database along a visibility continuum, we may find a reverse relation with audibility: the more invisible the database, the more present its sound. By this I do not argue in favor of neither loudness or quietness. I am only addressing the different possibilities that come from multiple access points to computers. Here I will use the words 'database' and 'computer' somewhat interchangeably. This decision comes from the fact, as I described in earlier sections, that computers cannot exist without databases. From this, we can further ask ourselves if all computer music is database music. As I hope to demonstrate, there are overt and covert uses of the database, but the database is ubiquitous in all computer practices (See Figure 1.8). The various disciplines at the intersection of music and computers take each a different approach to databases and, thus, to database performance. In this sense I describe and discuss the scope of actions that comprise database performance within three

practices using computers and sound: MIR, sonification, and computer music.

### 1.3.1 Music Information Retrieval

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.

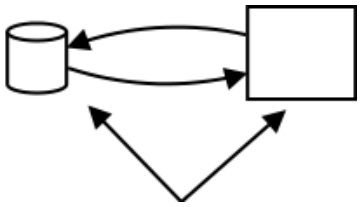


Figure 1.9: Diagram of database performance in MIR practices.
The database is visibly next to the computer, and the two bottom arrows indicate the intervention of the human operator.

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 pursposes. 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 1.3.3). Some examples of these notation databases can be the Polish folksong 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,[5] 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.[6] Some specific uses that MIR has given to databases have been:

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

[6] 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 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)
- 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).

### 1.3.2 Sonification

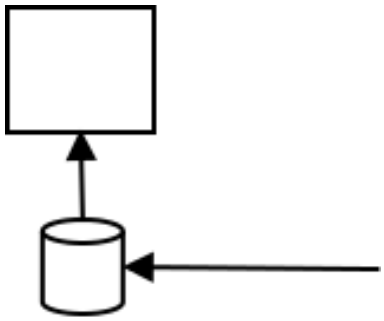


Figure 1.10: Diagram of database performance in sonification practices.
The database is visibly below the computer, and it feeds the computer from an external source represented by the right-most arrow.

The database is the ground floor of sonification. The sonified data is very likely to be digital,[7] 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: 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

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

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.

### 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.[8]

**Medical Images** Cádiz et al. (2015) proposed a sonification approach based on statistical descriptors of regions of interest (ROI) selected from medical images. In their study, they focused on

[8]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)

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.

### 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, navigable 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.

### Artistic sonification

**Wolves** J. Klein (1998) 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 (J. 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" (J. 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** B. L. 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** 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** 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 backround (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).

### 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), 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 highpass 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[9], 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 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*[10], they used the generated waveforms as a database, composing each waveform together with their visual generators as a collage.

**Ucanny Faces** Simonelli, Delgadino, and Cámara 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, 2018a).

### 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

[9] Apple's built-in framework to interface with the GPU. See https://developer.apple.com/documentation/metal
[10] https://vimeo.com/167646306

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: Pauletto and Hunt (2004a), Pauletto and Hunt (2004b), Lodha, Beahan, Joseph, and Zaneulman (1998), Beilharz and Ferguson (2009), Hildebrandt, Hermann, and Rinderle-Ma (2014), Worrall, Bylstra, Barrass, and Dean (2007), Walker and Cothran (2003), Vicinanza (2006)

### 1.3.3 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 of 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.

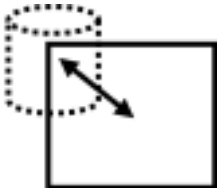


Figure 1.11: Diagram of database performance in computer music practices.
The database is invisibly behind the computer, within the softwares used to create musical works.

### 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 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 SSSP, he focused on Human Computer Interaction (HCI) —a field in its very early stages in 1978—[11]. 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

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

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 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, 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* (Xenakis, 1992), 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 1.3.3). 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 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 suggest 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 1.2.3) 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 is the possibility condition for 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 1.1.5).

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 inform, 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' 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 1.2.2), 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 simulaneous 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).

**Balance**

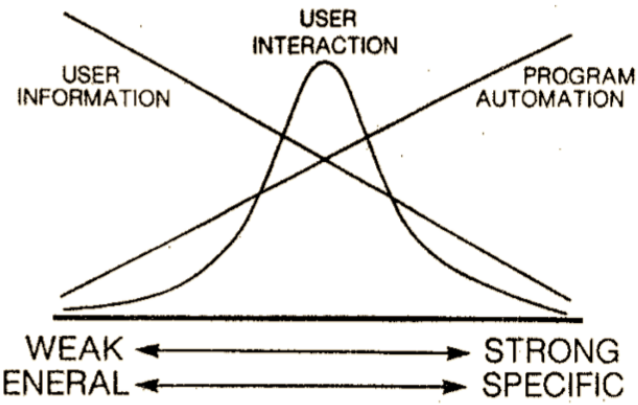



Fig. 1. Variation of user information, program automation, and user interaction as a function of the generality and strength of a system.

Figure 1.12: 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).

> ... 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 1.12), 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 structuresm 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.

### 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 Leeland 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).

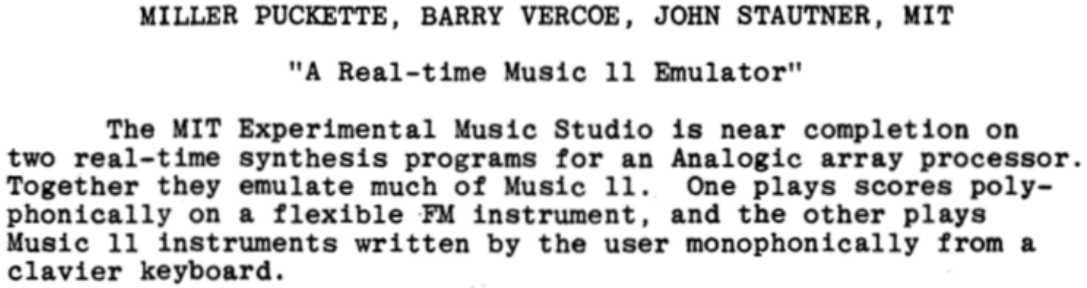
**From Staves to Speakers** Other programming approaches stemming from DARMS and SCORE were developed during the 1980s. Clements (1980) joined together the DARMS data structures with those used in MUSIC-V in a first attempt to obtain sonic feedback out of a notation system. Clements' attempt was nonetheless 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 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.

**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 1.13: A real-time version of MUSIC-11.
A bodyless 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 1.13). 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.[12] 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 Illioni's Computer Based Education Research Laboratory (CERL) (Scaletti, 1987). It is designed as an interactive composition

[12] "The scheduler always sends the first message in the lowest-latency nonempty 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).

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 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 "circunvent" (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.[13]

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

**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 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).[14] 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

[14]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 the SSSP's user-defined features for graphical representations.

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 accesible to the user. A common and elementary database routine (`sort`) that emerged to the program's surface because of traditional music notation practices.

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 (ibid.) (See 1.3.3), 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 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 hierarhcy 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 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, 2005a).

### 1.3.4 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.

### 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 (Schwarz, 2000; Schwarz, 2003; Schwarz, 2006a). By segmenting a large database of source sounds into units, a selection algorithim is used to find any given target by looking for "units that match best the sound or musical phrase to be synthesised" (Schwarz, 2006a). 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 refering to larger databases). That is to say, by joining together recorded samples, Scwharz ob-

tained 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 (Schwarz & Schnell, 2009; 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. Kobayashi (2003) 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. Collins (2007) 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" (ibid., p. 1). A recent case in which concatenative synthesis was applied to rhythm can be found in Nuanàin, Jordà, and Herrera (2016). Ariza (2003) was able to implement a model for heterophonic texture by pitch-tracking the highly ornamented music of the Csángó[15] 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* (Carpentier, Tardieu, Rodet, & Saint-James, 2006), 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

[15] "The Csángó, in some cases a Szekler ethnic group, are found in eastern Transylvania (Kalotaszeg), the Gyimes valley, and Moldavia" (Ariza, 2003).

as a function of its use. A list covering all extensions would require a 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. B. Sturm (2004) developed *MATCONCAT*, a concatenative synthesis library for *Matlab*. Schwarz (2006b) 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` (Brent, 2010). 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 ony one of them is concatenative synthesis.

### 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 (Vinet, Herrera, & Pachet, 2002a; Vinet, Herrera, & Pachet, 2002b; Vinet, 2005). 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. Norman and Amatriain (2007) enabled users generation of personalized audio description databases that could also be queried by content in *Data Jockey*.

**Similarity-based** Frisson (2015) 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* (Casey & Grierson, 2007) 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).[16]

Querying a database by similarity appeared in contexts other than performance workstations. Based on both the *Semantic Hi-Fi* and *SoundSpotter* projects, Price and Rebelo (2008) developed an installation with an interface to a relational database of percussive sounds.[17] 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 syntesis 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 many examples in the literature. Schloss, Driessen, Peter, and F. (2001) used audio analysis to obtain gesture features from the non-audio signals obtained from the *Radio Drum*, an instrument built at Bell Labs by

[16]"Audio Shingling is a technique for similarity matching that concatenates audio feature vectors into a sequence of vectors, and matches the entire sequence" (Casey & Grierson, 2007). "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" (Casey & Slaney, 2006).

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

Max Mathews in the late 1980s that uses a mallet as controller within three-dimensional space (Boie, Mathews, & Schloss, 1989). Schloss et al. (2001) searched for peak detection in the incoming signal to determine mallet (air) strokes. At CCRMA, Serafin et al. (2001) 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. Caramiaux, Bevilacqua, and Schnell (2011) 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 Cartwright and Pardo (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.

### 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*-dimentional 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** Choi (2000), Choi, Zheng, and Chen (2000) presented an interactive installation 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. Choi et al. 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” (ibid., p. 4). 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” (ibid., p. 3) 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” (ibid., p. 1).

**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 (Hamilton, 2006). 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” (ibid., p. 601). 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 throuth 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 (Nakamoto & Kuhara, 2007). By implementing a database system (MySQL) to store and retrieve vocal parts, Nakamoto and Kuhara 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 (ibid.). 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, Han, Kuchera-Morin, and Wright (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. Taylor, Allison, Conlin, Oh, and Holmes (2014) implemented centralized database systems to include user-defined interfaces to be saved and shared within their mobile device platform. Carter (2017) presented a work that gives each member of the audience their own instrument through their cellphones. 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 (Carter (ibid.)) 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.

### 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 publishig results like in research or academic communities. This is the case of the multiple datasets that exist.[18] In 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 1.3.1), 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 (Jones, Lagrange, & Schloss, 2007). Jones et al. 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 (Schmeder, 2009), 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 DBMS. The motivation behind these datasets, besides research is mostly to provide open access to any user with a computer and an Internet connection. Young and Deshmane (2007) created web-accessible databases of gestural and audio data concerning violin bow strokes. Hochenbaum, Kapur, and Wright (2010) developed a gestural and audio joint

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

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) (García, Vinceslas, Tubau, & Maestre, 2011)—, or radically plural: listening subjects asked to move as if creating a sound (Visi, Caramiaux, Mcloughlin, & Miranda, 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 (Sterne, 2012, p. 8). 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), JSON, YAML, or `.bib` (LaTeXbibliography 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 XML language (Bullock & Frisk, 2009). 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; Nymoen & Jensenius, 2011). 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 (Collins, 2006; Collins, Mclean, Rohrhuber, & Ward, 2003; Nilson, 2007; m zmölnig & Eckel, 2015). In this brief section, I would like to point to the work of Roberts, Wright, Kuchera-Morin, and Höllerer, 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 (Xambo, Roma, Lerch, Barthet, & Fazekas, 2018).

### 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 learninc. For example, in iteractive music systems (Rowe, 1992), in improvisation systems (Assayag, Dubnov, & Delerue, 1999; Bloch & Dubnov, 2008), to model Electronic dance music (EDM) patterns (Vogl & Knees, 2017), analog synthesizer parameter settings (Loviscach, 2008). Multimodal datasets have also been used in training (Schöner, Cooper, Douglas, & Gershenfeld, 1998). 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 1.14) 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.

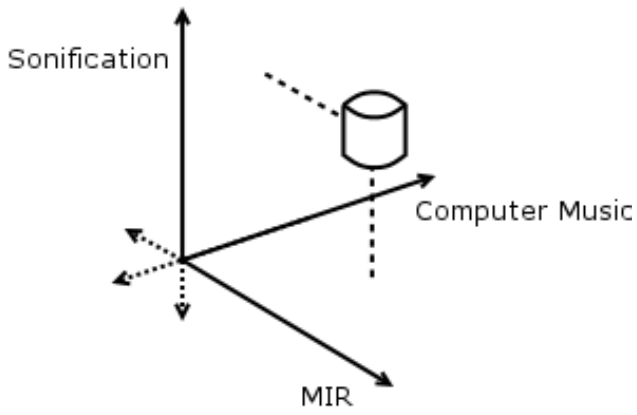



Figure 1.14: Intersection space.
Position of the database in terms of visibility among MIR, Sonification, and Computer Music.
Positive values indicate visibility and negative indicate invisibility.

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 just one frame. 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. This has been my task until now, and it is safe to say that we have looked at the database. In what follows, I will change gears and approach the database from a different perspective, one not guided by light, but immersed within sound.

# Chapter 2

# Database Aesthetics

## 2.1 Listening Databases

### 2.1.1 Interlude: I Am Sitting In A Room

The following transcription was made from a recording made at Lovely Music, Ltd (1981) of Alvin Lucier's work "I am sitting in a room" (Lucier, 1970). It is meant to indicate the many "irregularities" that differ from the text. The recording is available online.[1]

I am sitting in a room*mm*

diff~~e~~rent from the one you are in now

*inhale* (long)

I am re~~c~~$^{kh}$ording the ~~s~~$^{ss}$ound of my*hh* speaking ~~*voice*~~$^{voisss}$

*inhale* (short)

and I am going to play it . . . backin-to the room*mmm*

again*nnn* ~~*and*~~$^{an}$ again*n*

*inhale* (long)

un~~t~~$^{tx!}$il the

re~~s~~$^{ss}$onant*thh*→*ff* frequencies of-the-room

*inhale* (short)

[1] https://www.youtube.com/watch?v=fAxHlLK3Oyk

reinfor*c*~~*c*~~[sss] them*s*~~*s*~~[ss]elve*s*[sss]

*inhale* (long)

so that any

*sss*→emblance of my ~~*s*~~[sss]pee~~*ch*~~[ch→shwh]

*inhale* (long)

with per~~*h*~~[hhhh]ap~~*s*~~[s!] the excep~~*t*~~[shh]ion of

rwh[rhythm]

is*s* de~~*s*~~[ss]troye~~*d*~~[dh]

~~*W*~~[ou]*h*at you will hear-then*nn*

*inhale* (short)

are the

*nn*natural*l*

*rr*resonan*thf*→*ff*requencies-of-the-room*mm*

ar~~*t*~~[tzsh!]iculated by*ssss*speech*hhh*

I regard … this-~~*a*~~[a!]c-tivity

*inhale* (short)

nnnnn→ot *ss*so much*hhh* as-a-demon-~~*s*~~[ss]tra~~*t*~~[shh]ion of a physical f~~*a*~~[a→h]~~*c*~~[t→h]*t*

*inhale* (short)

*m*but more

*inhale* (short)

a~~*s*~~[z]a way to→ *S S SS SSss s sss s s*[smooth] out

any irregularitiesmy*hs*speech*hh* might hav~~*e*~~[f]

## 2.1.2 The Resonance Of A Return

**A Reverb** Sound reaches, enters, and traverses bodies in media. Media here refers to the matter through which sound propagates, such as a space filled with gas, liquid, or solid particles of matter including human and nonhuman bodies. More generally, sound propagation is conditioned by the qualities of the medium. Waves change direction by way of reflection or refraction, and they fade

out by way of attenuation. Furthermore, while the combination of density, pressure, temperature, and motion affect the speed of sound, a medium's viscosity affects the sound's attenuation rate. For instance, within hot and humid climates sound will move slower, or if there is wind blowing in the same direction of a sound, it will make the sound travel faster. This means that sound waves are affected in different ways by different media, some being more (concert halls) or less (anechoic chambers) resonant.

**A Filter** A listening body is part of the medium through which sound propagates: the body's sense perception is immersed within that medium. Sound, in its most basic and general form makes listeners vibrate as listeners become part of sound. Being part of sound, bodies change sound even before listening. On a mechanical level, we can think of the body as an a priori physical filter. Sound is filtered differently and uniquely within each body: my body changes the incoming sound for me, just as it does for others. In other words, a longitudinal wave passing through a body affects how it will arrive at other points in space. Therefore, bodies filter sounds for other bodies while affecting sound waves before they reach the listener. That is to say, since the listener's body itself refracts, reflects, and attenuates waves, the singular filter that is the body changes wave propagation not only for itself and its own listening experience, also for the listening experience of others. Empty concert halls are thus more reverberant than filled ones.

**A Loop** The filtering qualities of the listening body reveal the extent to which listening is such a singular and personal experience. Furthermore, this singularity can be understood as emerging out of the plurality that is sound. Plurality, in this sense, refers to the infinitesimal activity of waves. The interaction between the singular and the plural, in this sense, can be approached with the structure of a loop. I listen to myself as resonant subject, while creating meaning from a certain quality of a sound. I do this in simultaneity with others, who also create themselves as resonant subjects, while giving meaning to other sound waves. In this resonance, the vibrating link in

between ourselves is also simultaneously changing the way we are listening. Thus, every singular listening subject is in a state of being (mutually) (self) exposed to every other listening subject, that is, in resonance or in touch with one other.

**An Attack** Philosopher Jean-Luc Nancy (2007) brings forth an ontology of sound that can be understood in terms of resonance. He speaks about a "sonorous presence" (Gratton & Morin, 2015, pp. 143–144) that exposes listeners to themselves and to one another. The duration of this exposure is always an instant. All mechanical waves require an initial energy input and in the case of sound, particularly in musical contexts, this input is generally referred to as an attack. Instead, Nancy uses this term to describe the exact moment when a sound arrives and simultaneously leaves the body: the instantaneous appearance of sound within the body. An attack therefore instantiates the sonorous presence. Within this attack, that is, during the experience of this exposure, sound is understood as a sensing experience in itself as well as the experience of what a given sound might signify. As Brian Kane writes, to be listening in the sonorous presence constitutes "a mode of listening that exposes itself to sense" (ibid., pp. 143–144). This means that in the sonorous presence, the body begins to listen to itself listen. As I described with Hansen's notion of virtuality (See 1.1.6), virtuality can be understood as our brain's capacity to create images from the world. In listening, the virtuality of the human mind engages with the attack of the sonorous presence. In this sonorous present the first 'image' is the body itself. In this sense, the creative capacity of the body enables the body itself to be *self*-in-formed during the sonorous present. Furthermore, the body listening to itself listening results not only in the self-image of the body, it also creates an image of the listened.

**A Sampler** Consider, for example, an acousmatic concert in which one of the music works is made with pre-recorded violin samples. When this violin begins playing sounds, an illusion may very well begin to emerge: we can imagine a violin player. If the imaginary player continues to

play sounds and move them in space, this illusion continues in the direction of physical but illusory motion in space, that is, we can perceive an actual violin and an actual violin player. Therefore, this virtuality may project itself throughout the complete music work, thus grounding the music work on an imaginary force that is only alive because of the listener's own capacitiy for virtuality. The ghostly qualities of this force will be addressed further down this text (See 2.2.4). Most presently is the fact that this 'magic' show —happening in front and because of the listener's body-sensing mind— can be understood in terms of a resonant link between the human and the nonhuman: a web of interconnected objects that refer to each other. In this listening process, therefore, the listening subject exposes itself to itself and to the virtual self of the violin. Virtual, in this context, does not mean in opposition to the real. The virtual comes as the affective presence of a reality, and thus it becomes the possibility condition for the reality of images.

**A Texture** Since every 'body' is immersed within sound, and since sound refers to the thing that makes it, for Nancy this immersion is within a web of references. Furthermore, this web of references moves like waves: in time and space, back and forth, delaying in every next moment and distinguishing in every other reference. Therefore, instead of a loop, a more convoluted circuitry appears that can be understood as a Feedback Delay Network (FDN). The trick here is that this delay network sounds without input or output: it is already playing and sounding as a web-like endless texture. Brian Kane refers to this structure as "a structure of infinite referrals and deferrals" (ibid., p. 143), where references are at once postponed or delayed, and distinguished from each other. Within this 'texture,' Nancy approaches a notion of meaning: "meaning is made of a totality of referrals: from sign to a thing, from a state of things to a quality, from a subject to another subject or to itself, all simultaneously" (Nancy, 2007, pp. 4–9). Therefore, since sound "is also made of referrals: it spreads in space, where it resounds while still resounding 'in me'" (ibid., pp. 4–9), the result can be understood as a process that intertwines sense and signification. If this is the case, then sense refers to the body sensing itself sensing, and signification points to

the referential quality of the texture. In both cases, what is at stake in the listening experience is this interconnected web-like texture of delays and distinctions. On the one hand, the points in this texture are distributed in time, delayed to further moments. On the other, these same points are marks that indicate the extent to which they differ or ressemble each other, thus they can be understood as a spatial distribution of references.

**A Return** For Nancy, to be listening is to enter into "tension" and to be attentive for a relation to self (12 ibid., All subsequent quotes from this passage.). In this tension, 'self' refers neither to yourself —"not . . . a relationship to 'me' (the supposedly given subject)"—, nor to the self of another —"the 'self' of the other (the speaker, the musician, also supposedly given, with his subjectivity)." The structure of resonance can be understood in terms of a "relationship in self." That is to say, because of this relationship (in self) that appears in the play of the web-like texture of delays and references, to be listening is an ontological passage: "passing over to the register of presence to self." The self appears, it becomes present, as something that emerges from a resonant plurality. However, 'self' is not an expressive substance inherent to bodies, or already in the body, as if it were some originary essence that appears out of resonance. For Nancy, the 'self' is "nothing available (substantial or subsistent) to which one can be 'present.'" On the contrary, the self comes in the form of a return, the "resonance of a return [*renvoi*]"

The more general implications of this ontology would extend the limits of this dissertation. For a commentary on Nancy's work, see Gratton and Morin (2015). Nevertheless, I would like to point to one particularity of this ontology of sound: listening is an activity of sensing bodies through which their ontological condition becomes available. In this sense, to what extent can we consider the database as a listening body? And if so, to what extent is there an ontology of the database? These are the questions that I address during the following sections.

### 2.1.3 Resonant Network

**The Recorded Movement of a Thing** Philosopher Bruno Latour (1990, 1993) developed a theory of networks called . can be understood as a way of connecting and associating entities to one another. It is a tool that builds an image of the world made of nodes along a decentralized web of meaning. Latour reformulates nodes and edges, with what he calls 'semiotic actors,' 'actants,' or 'agents' (nodes) and of the interconnected accounts that these have of each other (edges). As I have mentioned earlier with the network model in databases (See 1.2.4), navigating through networks is traversing from node to node. However, the (visual) two-dimentional metaphor of a 'network' as a 'surface' limits the understanding of its topology: "instead of surfaces one gets filaments." (Latour, 1990, p. 3) In this sense, he points to a misunderstanding that comes from giving a technical definition such as the one described with the database model: "nothing is more intensely connected, more distant, more compulsory and more strategically organized than a computer network" (ibid., p. 2). points towards a topological shift. Nevertheless, the navigational paradigm advanced by Bachman (1973) in relation to databases whose 'keys' become n-dimensional space, does translate well to Latour's model, because nodes have "as many dimensions as they have connections" (Latour, 1990, p. 3). In any case, what this navigation points to is that is comprised entirely of motion and activity: "no net exists independently of the very act of tracing it, and no tracing is done by an actor exterior to the net" (ibid., p. 14). Thus, meaning and connectivity are enabled by the activity or work of actors: "In order to explain, to account, to observe, to prove, to argue, to dominate and to see, [an observer] has to move around and work, I should say it has to 'network"' (ibid., p. 13). This work, *net*-work, or 'tracing,' is not only the movement of associations and connections, it is also the 'recording' of this movement. In this sense, Latour claims that "a network is not a thing but the recorded movement of a thing" (ibid., p. 14). Furthermore, nothing falls outside the network: "the surface 'in between' networks is either connected —but then the network is expanding— or non-existing. Literally, a network has no outside (ibid., p. 6)." The network encompasses its own

actors and its own expansive motion. Most importantly,  is a tool aimed at describing the nature of society. However, in this description,  "does not limit itself to human individual actors but extend[s] the word actor. . . to non-human, non individual entities" (ibid., p. 2).

**Howling** Thinking networks in terms of a (sonic) three-dimentional metaphor (as a mechanical wave) is thus misunderstanding it. Coupling 'resonant' and 'network' results in a sort of (impossible) positive feedback. While a network expands in redundancy, overflow, accumulation, and self-reference, a sound attenuates towards imperceptible and infinitesimal thresholds. Lending an ear to the sound of  we would find ourselves listening to expanding filaments. However, as an acoustic experiment that would combine the circuitry of a feedback with the accumulative quality of networks, I propose to consider Lucier's (1970) "I am sitting in a room". I have transcribed this sound art piece at the beginning of this chapter, as it is self explanatory (See 2.1.1). It can be understood as a triple crossfade, first between speech and music, gradually crossfading into a second crossfade, between timbre and space. Through the circuitry of a closed and controlled feedback loop between a microphone, a speaker, and a room. More considerations of this work I will leave for some other time, and I will refer to Valle and Sanfilippo (2012) for further readings on feedback systems. What I would like to bring here is an experimental revision of what is known as the Larsen effect: "in every sound reinforcement system, where the loudspeaker and the microphone are placed in the same acoustic environment, a certain amount of the signal radiated by the loudspeaker is fed back into the microphone" (Kroher, 2011, p. 11). When these systems become unstable, the Larsen effect appears (also refered to as 'howling'), "resulting in a positive feedback producing pitched tones from the iterated amplification of a signal" (Valle & Sanfilippo, 2012, p. 31). Therefore, in Lucier's room, what occurs is quite literally the Larsen phenomenon, but "stretched in time," and thus the "room itself acts like a filter" (ibid., p. 34). Considering the mechanical contradiction in thinking resonant networks, I believe it necessary, then, to expand the 'mechanical' side of the feedback system in question: Lucier's *room* needs to be expanding as

well. As a consequence, the "resonant frequencies" (nodes) of the expanding network would cease to "reinforce themselves." However, (and here is the experiment) this does not mean that these nodes would cease to act, let alone resonate. In this sense, we can ask ourselves how would *this* sound like? *Where* would the 'I' be actually *sitting*?

**The Resonant Movement of a Thing** Such a feedback network would redefine the notion of a temporal delay into a *spatial* delay. Instead of the Larsen effect being "spread in time," in Lucier's work it would also spread through space. The room as a filter would resonate differently because it would be understood as a texture, a networked resonance. If Latour's semiotic actors are in constant reference to each other, it can be argued that they are in resonance with each other, in a permanent state of vibration, or simply, *listening*. Thus, Latour's phrase can (perhaps) be reformulated: *the network is not a thing, but the resonant movement of a thing*.

**I am sitting in a database...** This is the crucial leap that comes out of the idea of a resonant network: the moment the nonhuman in the network is comprehended as resonant, it is the moment that they engage with an approach to self (in Nancy's terms). Following this logical thread, a database can be considered as a semiotic actor as well as a resonant subject. On one hand, databases are not networks, they are agents: acting, tracing, and listening within a network. On the other hand, since databases are indeed listening, to what extent can we think of them as listening to themselves listening? Bringing back Lucier and his room, I would like to address this question with another aspect of the work. (And by 'work' I begin to introduce an important aspect of this dissertation, a concept that embraces activity, productivity, but also product, and objects: operativity and opus.) There is indeed a fourth crossfade, between the 'I' in the text, and the 'I' in the voice that reads it. The simplest way to approach this is by asking ourselves, if after recording the first input signal Lucier remained seated *in* the room or not. This is a difference that cannot be approached from the recording itself because it is inaudible. I will refer to this difference further down this text. For now

I point to the fact that the moment Lucier recorded his voice, the 'I' began residing in the loop. I believe this is one of the most crucial 'irregularities' that can be found throughout the work. In the interlude at the beginning of this chapter, I transcribe the text as Lucier reads it. I attempted to be as clear as possible, crossing out, replacing, extending all the consonants into what I thought was a more faithful score for the read fragment. Thinking as a composer, this score (with all its notated irregularities) would explain the first minutes so fiercefully that the mystery of the last minutes would be solved. But, the most crucial aspect of the piece cannot be rendered in symbolic transcription, because the 'I' is somewhere in between the transcribed and the inscribed (See 2.2.4). This 'I' is what is at stake when databases begin to resonate, that is, it is the approach to this notion of 'self' what begins to redefine ourselves in general. That is to say, within the resonant network we face a 'self' that changes our own notion of 'self' in general. In this sense, a 'self' *sitting in a database* returns to us (resonates back) putting into question a relationship (a difference): what is the difference between the two 'I's? Is is this same difference at stake between the human and the nonhumam? The implications of these question I will move forth in the remaining sections of this dissertation. However, the most present step is analyzing the conjunction that the two clauses of the question points to: the sharing of the 'I,' an exposure of community.

### 2.1.4 The Unworking Network

**Community as unwork** As I have described above, for Nancy, a resonant self (the 'self' from now on) is made of "the singular occurrences of a state, a tension, or, precisely, a 'sense"' (Nancy, 2007, p. 8). Since these occurrences are in a permanent state of occurring, that is, never fixated in a whole (in 'tension'), we can understand singular beings as being 'interrupted' or 'suspended.' These descriptions come from an earlier text Nancy (1991), in which he extends this definition of the self to the definition of a community: "community is made of the interruption of singularities, or of the suspension that singular beings are" (31 ibid., All subsequent quotes from this passage.).

This is why we can understsand 'community' as ontological (grounded on a nature of being), and not as teleological (grounded on a purpose or an objective). In other words, if community is thought of as a product of the work of 'selves' (teleologically), it follows that selves could also become the product of community. Nancy thus underscores that "community is not the work of singular beings, nor can it claim them as its works" (See Figure 2.1). Since community is ontological, Nancy understands it as the being of singular beings "suspended upon its limit." Therefore, within this ontology of community, how is it possible for us to speak of the 'work' of 'selves' and of community if 'work' is something that does not enter into its definition? Furhtermore, how is the concept of the work of art redefined or framed within this ontology? Nancy's conclusion is that community can never result out of 'work,' but it is something that unfolds as 'unworking.' Borrowing from Maurice Blanchot's concept of *desoevrement*, Nancy proposes 'unworking' or 'inoperativity' as a way to understand 'work' within an ontology of community. 'Unwork' is work withdrawing from itself: "that which before or beyond the work, withdraws from the work." That is to say, Nancy points to a moment in which operativity separates from itself, and by that gesture, it distinguishes itself from both production and from a whole, or a finished product: "no longer having to do either with production or with completion, encounters interruption, fragmentation, suspension." Is this not the resonance of a return? Work returning as the interrupted resonance of its own unworking?

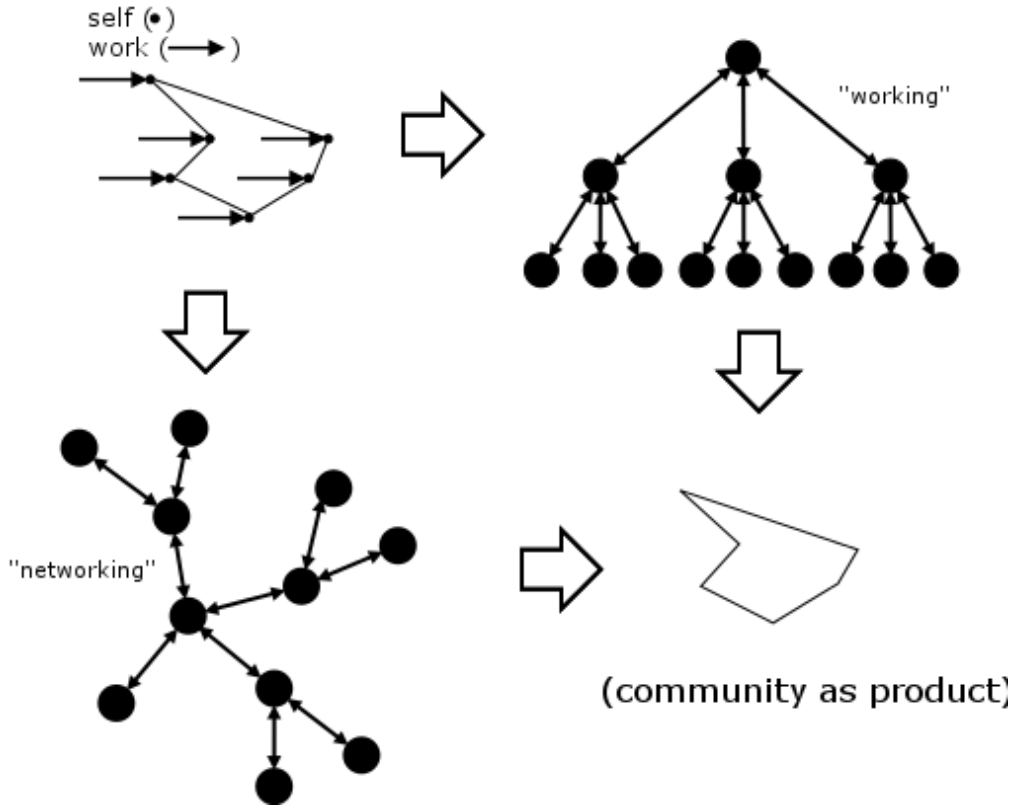



Figure 2.1: Community as work
A graph displaying the teleology of the work (arrows) of selves (dots) in community (dots joined by arrows), in relation to the first two models with which databases were designed: hierarchical (right) and network (left).

**At the Limit** Nancy's concept of community can be recognized within his later and broader concept of 'resonance.' Given the fact that Nancy's ontology of sound points to the distance or the interval between sense and signification, and thus, to the emergence of a resonant subject during the sonorous presence, this distance can be thought of as suspended at a limit. We can think of this limit as an edge in the resonant network that, in Nancy's terms, exposes selves to themselves and to one another. Further, Nancy provides us with an essential insight, suggesting that "it is not obvious that the community of singularities is limited to 'man' and excludes, for example, the 'animal."' (ibid., p. 28) Therefore, by understanding resonant networks in terms of community we can speak of an exposure of selves, or a self-exposure, between humans and nonhumans in a liminality can be thought of as a skin. Not in the sense of a layer that separates the interior form the exterior of a body, or, for that matter, as a surface under which or over which two selves can connect. For this would would make us fall out of Latour's definition of the network. This skin is not a surface, but it can be thought of as a texture; not a layer, but an interweaving of minuscular threads that, in their own locality are fragile, but due to their reticulated structure expand into a redundancy of fragilities that prevents concepts such as unity, concentration, or purity, to enter into the picture. Within this quality of incompleteness, which relates to suspension, but also to fracture, instability, and unpredictability, is how the symmetry of this teleology of work also relates to communication. Understood as the being in common of singularities: "communication is the unworking of work that is social, economic, technical, and institutional (ibid., p. 31). I am tempted here to suggest that it communication, as well as community, is also the unworking of work that is resonant.

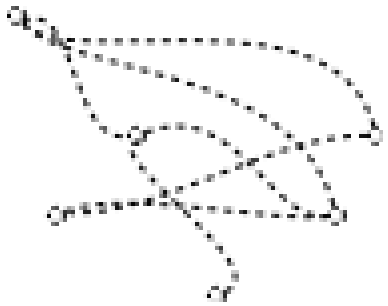


Figure 2.2: Community as unwork
A graph displaying community as an ontology of unwork.

(A process which composes itself as process),

In the network model, the nodes within the network are connected and associated to each other, that is, the 'work' of agents in Latour's terminology. As I describe earlier, Latour's network encompasses the human and the nonhuman:

Agency, therefore, is what is in common across all nodes. However, a crucial distinction appears if we understand this agency as a form of 'networking,' that is, as the way this web of associations and connections works: In other words, consider the graph that I include here (See Figure 2.1).

"extend the word actor -or actant- to non-human, non individual entities" (Latour, 1990, p. 2).

"The notion of network allows us to lift the tyranny of social theorists and to regain some margin of manoeuvers between the ingredients of society —its vertical space, its hierarchy, its layering, its macro scale, its wholeness, its overarching character— and how these features are achieved and which stuff they are made of" (ibid., p. 5).

"how these features are achieved and which stuff they are made of.The notion of network, in its barest topological outline, allows us already to reshuffle spatial metaphors that have rendered the study of society-nature so difficult: close and far, up and down, local and global, inside and outside. They are replaced by associations and connections (which AT does not have to qualify as

being either social or natural or technical as I will show below) . This is not to say that there is nothing like "macro" society, or "outside" nature as the AT is often accused to, but that in order to obtain the effects of distance, proximity, hierarchies, connectedness, outsiderness and surfaces, an enormous supplementary work has to be done. This work however is not captured by the topological notion of network no matter how sophisticated we wish to make it." (ibid., p. 6)

". An actor-network is an entity that does the tracing and the inscribing. It is an ontological definition and not a piece of inert matter in the hands of others, especially of human planners or designers. It is in order to point out this essential feature that the word "actor" was added to it. (ibid., p. 7)

HYBRIDS: extending semiotics to things instead of limiting it to meaning

"What really matters is that it is an elevation instead of a reduction and that the new hybrid status give to all entities both the action, variety and circulating existence recognized in the study of textual characters and also the reality, solidity, externality that was recognized in things "out of" our representations. What is lost is the absolute distinction between representation and things -but such is exactly what AT wishes to redistribute through what I call a counter-copernican revolution" (ibid., p. 11).

**Database Community** The resistance of the skin represents, thus, the resistance of connectivity itself, that is, the resistance of a community.

In the resonant network model I am proposing here database performers (or databasers) are human actors in the network, while nonhuman actors include the database.

The resonant network, as an instantiation of a process of unworking, enables the thought of a database community to emerge. This community consists of .....

Therefore, by substituting, on one hand, the human with the database performer (databaser) and, on the other, the nonhuman with the database, the resonant network, as an instantiation of a process of unworking, enables the thought of a *database community* to emerge.

In a database community, database music can be understood as a hybridly social and communicative event.

This is to say that, following Latour's hybridity of objects in his understanding of society (ibid., p. 2), database music is a social practice that comes as a result of databasers and databases in resonance with each other.

Furthermore, database music is an instance of community,

in the sense that it is an event of communication, understanding the communicative as "the unworking of work that is social, economic, technical, and institutional" (Nancy, 1991, p. 31).

Therefore, the database community emerges as the unworking of databasers and databases.

Database community can be considered as, precisely, the skin

upon which the human and the nonhuman resonate, which amounts to —but never finalizes in— a musical event.

## 2.2 Databases And Memory

### 2.2.1 Interlude: Embodied Memory

> I suspect, nevertheless, that he was not very capable of thought. To think is to forget differences, to generalize, to abstract. (Borges, 1942, p. 2)

The importance of memory —and forgetfulness— can be represented by Jorge Luis Borges's famous 1942 short story, *Funes, the memorious* (ibid.). Due to an unfortunate accident, the young Irineo Funes was —"blessed or cursed" as Hayles points out (Hayles, 1993, p. 156)— with an ability to "remember every sensation and thought in all its particularity and uniqueness " (ibid.). A blessing, since a capacity to remember with great detail is certainly a virtue and a useful resource for life in general; a curse, because he was unable to forget and, as a consequence, he was unable to think, to remember, to dream, to imagine. Throughout the years, he became condemned

to absolute memory, and so to its consequence, insomnia:[2] he was secluded in a dark and enclosed space so as not to perceive the world.[3] Hayles focuses on one aspect of the story, namely, the fact that Funes invented —and begun performing— the infinite task of naming all integers, that is, of giving a unique name —and sometimes, last name— to each number without any sequential reference. According to how Hayles describes it, by carrying out his number scheme, Funes epitomizes the impossibilities that disembodiment brings forth. As Hayles writes, "if embodiment could be articulated separately from the body ... it would be like Funes's numbers, *a froth of discrete utterances registering the continuous and infinite play of difference*" [emphasis added] (ibid., pp. 156–159).

Therefore, the point that she is touching is that of the limits and fragility of embodied memory. In that Manovichian world which "appears to us as an endless and unstructured collection of images, texts, and other data records" (Manovich, 2001, p. 219), this idea would be perfectly viable. Indeed, data banks have already been growing exponentially much in the same way as Borges' 1942 character's mind was aiming at. This capability of accumulation without the need of erasure is enabled by the database structure inherent in computers.[4] However, the distinction that Hayles presents —which has been discussed before (See 1.1.4)— is crucial: data is not information because information needs to be embodied. Therefore, on one hand, a disembodied data bank can have all the uniqueness and difference that is available by the sum of all cloud computing and storage to date; however, on the other hand, an embodied memory is only available by the human capacity to forget.

Matías Borg Oviedo (Oviedo, 2019) relates this incapacity for thought precisely to the negation of narrativity itself, thus finding in the image of Funes a hyperbole for contemporary sub-

[2] In the prologue to *Ficciones*, Borges writes that this story is a long metaphor of insomnia: "Una larga metáfora del insomnio" (Oviedo, 2019).

[3] Within this fictional universe, the only way for him to sleep was to imagine the opaqueness of an unknowable future...

[4] In fact, the demand when it comes to computers is less its ability to erase —or even compress data— than storage space, a hardware-dependent commodity that has circulated ever since Von Neumann's architecture came into the picture (See 1.2.2).

jectivity (ibid., p. 5), where there is no room for narration, only accumulation of data. In this sense, narrativity can be seen as that which resides in the threshold between knowledge (i.e., memory) and storage (i.e., archives, databases). I believe this distinction stems precisely from the difference between information and data. The process of information, of giving form, requires a certain temporality that is not that of the immediate and extremely operative zero-time of the (computer) processor. Within the zero-time of computer operations, there simply is no time for narrative, only for addition, for an increment. With this in mind, I would like to question Manovich's opposition of narrative and database, precisely on the grounds that narrative is temporal —happening as a historical process— and algorithms are atemporal —operating in an effervescent now. Therefore, Funes' accumulative memory represents the overflow of the now that precludes narration: neither data structures nor algorithms can forget to count.

Studies in cognitive psychology might have something to add here. In Wessel and Moulds' commentary (Wessel & Moulds, 2008) on Paul Connerton's *Seven Types of Forgetting*, the authors consider forgetting to be the "failure" of certain search processes on account of an inability to recall information from memory. While pointing to current psychological models of memory which consider forgetting to be an adaptive and functional activity, the authors acknowledge the mystery of certain aspects of memory: "In human memory, it is unclear what really happens to old, disused or deliberately ignored memory traces —they might be retrievable, they might be lost, but no-one can tell" (ibid., p. 292).

Thus, considering this distinction between embodied and disembodied memory, I propose an imaginary experiment, one that I believe was missing from Borges' short story, however utterly fantastic his writing was.

One thing that can be read from the story is that, in order to seclude himself from perceiving the world, or better, in order to forget the world altogether, Irineo stayed in the dark. This is how he cancelled light, a quite powerful stimuli if memory-space is to be optimized —for the pur-

pose of, say, getting some sleep.[5] However, there is little to no mention of the sonic environment in which Funes was embedded —probably in the outskirts of the quiet Uruguayan city of Fray Bentos. In fact, the only sonic references are focused on the narrator's perspective, referring to Funes' high-pitched and —due to his being in the darkness— acousmatic voice.[6] Therefore, focusing on Funes' listening, by locking himself inside a room he would have managed to attenuate sound waves coming in from outside. Notwithstanding his isolation —or, better, his self-imprisonment—, sound waves are actually very difficult to cancel. An interesting experiment would have been to have John Cage take Irineo to an anechoic chamber and ask him what he can remember then. From Cage's own experience, we can guess that Funes would effectively remember his own sounding body.

(It is interesting to compare Funes' search for filtering out the world with John Cage's search for silence. Kim Cascone writes that "[Cage's] experience in an anechoic chamber at Harvard University prior to composing 4'33" shattered the belief that silence was obtainable and revealed that the state of 'nothing' was a condition filled with everything we filtered out" (Cascone, 2000, p. 14). It is interesting to place an 80 year-old Irineo in David Tudor's premiere at Maverick Concert Hall in Woodstock, NY, infinitely listening to 4'33")

However, it is very unlikely —but nonetheless possible— that Borges was aware of American acoustician Leo Beranek's research for the US Army during World War II, that is, when the first anechoic chamber was built.[7] Furthermore, even if he managed to isolate himself perfectly from the world, cancelling perception altogether, Funes would have been with his memories, which are not discrete, but continuous iterations of the world he had accumulated over the years. What this means is that all the sounds he had listened to would be available to his imagination.

As far as we can learn from the narrator, while *smell* is referenced to in the story, *sound*

[5]For example, one of Irineo's concerns was to reduce the amount of memories on a single day, which he downsized to about seventy thousand...

[6]This acousmatic quality of Funes' voice will not be touched here, but it is indeed a good point of departure for an essay.

[7]https://en.wikipedia.org/wiki/Leo_Beranek

was completely out of Funes' concerns. Therefore, the question is how would the world sound for Irineo Funes? The task is not difficult to imagine: the world would be inscribed in poor Irineo's memory in such an infinitely continuous way that each fraction of wave oscillation would be different, unique, leaving no space for repetition of any kind. All sounds would be listened completely, with every infinitesimal fraction of oscillation of the waves pointing to the most utterly complete scope of imaginable references. It would not be inaccurate to compare this type of memory saturation with CPU saturation, for example, the way a computer would —even the most gigantic multi-core imagined—, if it was commanded to compute, with accuracy, the wave equation. In this complete state of listening, there would be no possibility for thought, no processing of any kind, only infinite accumulation and storage.

In this sense, an infinitesimal incorporation of sound is unthinkable. This is not to be confused with the infinite structure of referrals and deferrals that Nancy's resonance points to. The problem here —and this is evident in the story itself— is in the intersection of the finite with the infinite. While the structure of sound itself is infinite, in the sense that it is an ongoing process —i.e., circular loop— of delays and references, the singularity of the listening subject is finite: its limit exists as its possibility condition. This is to say that, given such an ontology of sound, the emergence of a resonant subject occurs *at* the limit, that is, at the liminality of an exposure to itself, to others, to waves, etc. In the case of the story, while Funes' nonhuman qualities correspond to dynamics of the infinite, his human body is just like any other that was thrown in the world. Therefore, no matter what the narrator has the reader believe, Funes is not deprived of this liminality. Throughout the story, Funes' liminality grows more and more evidently, all the way until the end —an ending that, despite all his nonhumanly infinite qualities, is, nonetheless —be it a blessing or a curse—, utterly human.

Less than focusing on literary analysis, in this interlude I take the concept of an absolute memory within a human being to be at an intersection between disembodied theories of information and, precisely, the concept of an embodied memory. The aim is to differentiate between

human and nonhuman in terms of memory and databases, so as to provide a link between Latour's 'recorded movement' of the network, and Derrida's concept of the archive.

### 2.2.2 The Effraction Of The Trace

> Lo cierto es que vivimos postergando todo lo postergable; tal vez todos sabemos profundamente que somos inmortales y que tarde o temprano, todo hombre hará todas las cosas y sabrá todo.[8] (Borges, 1942)

> All these differences in the production of the trace may be reinterpreted as moments of deferring... Is it not already death at the origin of life which can defend itself against death only through an *economy* of death, through deferment, repetition, reserve? (Derrida, 1978, p. 202)

I would like to take a brief, but necessary, psychoanalytic detour, so as to set some context for the concept of memory. For this purpose, I take Derrida's reading of Freud's conceptualization of memory as a starting point towards distinguishing between human and nonhuman memory.

**Memory as Breaching** According to Derrida (ibid.), Freud understood memory as the essence of the psyche: "Memory, thus, is not a psychical property among others; it is the very essence of the psyche: resistance, and precisely, thereby, an opening to the effraction of the trace" (ibid., p. 201). In this sense, the perceived world enters as a force of effraction into the resisting unconscious, resulting in the inscription of a trace, that is, a memory. By definition, this is a violent process of impression (pressing *in*), which is only possible by the notion of resistance. Thus, in order for something to leave a mark, there has to be some force acting against an other. In this case, one force is the world with its constantly changing images; another, the unconscious with its resistance to change. In this sense, Freud describes memory as breaching, or better, a state of being opened to this effraction. Memories are inscribed with incredible violence, one that can define the extent

[8] Anthony Kerrigan translated this fragment as: "The truth is that we all live by leaving behind; no doubt we all profoundly know that we are immortal and that sooner or later every man will do all things and know everything." In a more literal translation of the first sentence, it reads: "What is certain is that we live deferring all that can be deferred."

to which the inscription can be recalled later on, for instance, the case of repression, which is not forgetfulness but the deferred case of a force of containment. Furthermore, this breaching can be equally thought of as a fracture, hence the notion of effraction, which speaks of a the violent rupture. Despite, however, this violent image of path-breaking, Derrida writes of memory as an opening, in a paradoxical game between the resistance of the unconscious and the unavoidable (unclosing) state of being innocently ready for the next perception.

**Breaching and *différance*** Derrida points to one issue in Freud's opposing forces: the case where resisting forces meet equally strong forces, which would result in a paralysis of memory. Therefore, he proposes that it is "the difference between breaches which is the true origin of memory, and thus of the psyche" (ibid., p. 201). In other words, Derrida reconfigures the psyche with the interplay of *différance*, that is, a structure of infinite referrals and deferrals: "trace as memory is not a pure breaching that might be reappropriated at any time as simple presence; it is rather the ungraspable and invisible difference between breaches" (ibid., p. 201) Therefore, since the essence of the psyche (memory) is breaching (tracing) and the space and temporal dislocation of traces (*différance*), a link between human and nonhuman can be established: writing.

**Hypomnesis and the Mystic Pad** As Derrida points out, writing as *hypomnesis* (as an externalization of memory) has been considered since Plato (ibid., p. 221). However, Freud's metaphor for the perceptual system found yet another place: the Mystic Pad. There were many derivations of this device, but it consists in something like a writable surface board. When you write on it, a thin layer on top sticks to a sort of wax on the back, thus leaving a trace; when you lift the upper layer, the traces vanish completely. This special device came to represent, for Freud, the structure of the perceptual apparatus itself, or as Derrida says, it is where "the psychical is caught up in an apparatus, and what is written will be more readily represented as a part extracted from the apparatus and 'materialized"' (ibid., p. 222). Therefore, this new (1925) hypomnesic device allowed Freud

to shift from considering regular writable surfaces (paper), to a combination of resisting textures in a device which allowed for "a perpetually available innocence and an infinite reserve of traces" (ibid., p. 223). Derrida, however, reconceptualized this differently. The crucial distinction he finds on this writing device is the notion of erasure, but most pressingly, that of repetition. In this sense, traces "produce the space of their inscription only by acceding to the period of their erasure" (ibid., p. 226). This means that tracing constitutes a process whose only possibility is its own negation, that is, its own undoing of itself. Furthermore, these traces are, from the first moment, that is, from the beginning of the process of impression, "constituted by the double force of repetition and erasure, legibility and illegibility" (ibid., p. 226). Hence, the spatio-temporal duality of the concept of *différance*, that is, the interplay between spacing and deferments, which comes to be the "work of memory" itself (ibid., p. 226).

> The 'subject' of writing does not exist if we mean by that some sovereign solitude of the author. *The subject of writing is a system of relations between strata*: the Mystic Pad, the psyche, society, the world. Within that scene, on that stage, the punctual simplicity of the classical subject is not to be found. In order to describe the structure, it is not enough to recall that one always writes for someone; and the oppositions sender-receiver, code-message, etc., remain extremely coarse instruments. We would search the 'public' in vain for the first reader: i.e., the first author of a work. [emphasis added] (ibid., p. 227)

**Nonhuman Authors** Since the structure of memory, as I outlined above, can be comprehended as path breaking and *différance*, the concepts of resistance, referrals, and deferrals play an important role in its definition. Furthermore, the externalization of memory comes as an instantiation of the very structure of not only our perceptual machine, but also of the structure of the psyche itself. This hypomnesis allows for a materialization of the psyche that is constituted as a tool, that is, as an apparatus to be handled. With this conceptualization of memory as writing, Derrida reconfigures the notion of the self of writing, that is, of an author. Therefore, when memory is thought of as writing, the classical notion of self begins to disappear, opening up the space for the nonhuman.

In what sense can this disappearance of the self be accounted for if we substitute writing for databasing? How does the databasing self emerges in this non-origin of the originary moment of writing that Derrida describes above? How is this "system of relations between strata" to be understood in terms of the work of databasers? This is how a further step into the conceptualization of memory can be of aid, one that considers memory as resonance, and writing as databasing. I have already described above how resonance and memory constitute processes of *différance*, that is, how the structure of sound, on the one hand, and traces, on the other, relate to spatiality and temporality. This is to say that, while the infinite situations of reference create a space of multiple situations, connections, associations, etc., which constitute an instance of signification, simultaneously, the ongoing process of deferral creates the oscillatory condition of perception, which constitutes the infinite return, the repetition, and the reverberation of a self. This self, however, is not an essential one, that is, it is not a substance, but it is understood as a resonant subject, *the resonance of a return*, which, like Derrida's "subject of writing", does not exist in itself, and only appears as a result of a resonant "system of relations between strata."

**Database as Agents** Therefore, if this resonant self can only emerge out of this system —or better, network— of relations, then, every resonant point of the network must be considered as an agent in the constitution of a self. This means not only that the database, in this particular case of database music, becomes an agent of self-hood —and, for that matter, an agent of authorship relating to the resulting work of database music. It also means that, as an instance of hypomnesis, that is, as technology that externalizes memory, the database appropriates the qualities relating to memory itself that were described above: breaching and *différance*. Thus, not only can the nonhuman be reconceptualized within these qualities, also the human itself becomes reconfigured when faced upon the situation that memory, as breaching and *différance* can also be found in the nonhuman. Therefore, given that the nonhuman and the human are engaged in this resonant network that cancels the classical notion of self, then, distinguishing between the human and the

nonhuman cannot be carried out in classical terms, that is, in terms of substance, essence, etc., since they do not apply anymore. The only thing that remains is, basically, the body, the singular instance of a resonant skin, whether it be of a human or a nonhuman, of a databaser or a database; a skin, but also an open resistance to the forces of change.

### 2.2.3 The Archontic Principle

**Archives and Memory** For Derrida, the archive cannot be reduced to memory, it is is neither a form of rememoration nor a "conscious reserve," but simply a place where "the origin speaks by itself" (Derrida & Prenowitz, 1995, p. 60). In this definition, the word 'origin' refers to the greek root of the word 'archive' [arkhē], whose meaning is twofold: on one hand, its definition is topological (space) on the other, it is nomological (law). In the topological sense, *arkhē* is related to the greek word arkheion, which, in turn, refers to "a house, a domicile, an address, the residence of the superior magistrates, the *archons*, those who commanded" (ibid., p. 9). In the nomological sense, *arkhē* refers to the ruling, to authority, to the command or commandment, i.e., the law. Thus, the magistrates [archons] are those who "have the power to interpret the archives," and they indeed reside inside this place called *archeion* (ibid., p. 9). Therefore, considering the fact that these magistrates actually reside in place of the archive, Derrida refers to an "institutional passage from the private to the public" which is constitutive of the archive itself (ibid., p. 9).

**Hierarchies** From this definition of the archive, what is alluded is the hierarchical structure of civilization itself, that is, of government and legislation. Going further into this aspect of the concept of the archive would extend the limits of this text. Derrida, with this conceptualization of the archive, focused on the exponential growth of archives that spawned during the 20th century, defining it as an impulse, or better, as a 'fever' and a drive. Thus, he proceeded to perform a psychoanalysis of this symptomatic condition of mid-1990 society, beginning with the archivization of the house of the *father* of psychoanalysis, Sigmund Freud. Therefore, I would like to focus on

one principle that Derrida described as belonging to the concept of the archive, what he calls the *archontic* principle.

**Archontic Principle** The archontic principle is a type of authority that the archive projects, which can be understood as powerful dictating rule that is before anything else. Hence, its categorization as principle, which is also related to the origin (e.g., the latin root *principium* which refers to the beginning) and to the figure of the ruler (e.g., principal, prince, etc.). For Derrida, this principle is constitutive of all words deriving from *archē*, which is why he understands the archive, for example, as being patriarchic. Since the archontic principle is present in the etymology of the word 'patriarchy' —which refers, likewise, to 'lineage,' 'father,' and 'I rule'—, therefore, he argues that the archontic is embedded first with a sense of filiation, that is, with fatherhood and the relationship between father and child. Thus, the archontic is "paternal and patriarchic" (ibid., p. 60).

**Patriarchy** What follows is that this patriarchy is grounded in a domicile (house) or an institution (family). Within this domiciliation or institutionalization is from where rules are prescribed, and where the ruling takes place. In the case of the patriarchic concept of family, the father is the ruler; in the case of civilization, the governor is the ruler. Therefore, this is what Derrida means when he performs a psychoanalysis of the archive: the topo-nomological aspect of the archive comes precisely from a —strategically male-centered— concept of fatherhood. In this sense, Derrida discovers that the archontic has the form of Freud's oedipus complex. An oedipus complex constitutes a desire of the child's unconscious hatred towards a parent. It is itself based in Sophocles' drama *Oedipus Rex*, in which such desire results in eventual parricide. Derrida notes that this complex has the form of an infinite loop between father and child, which is bound to repeat itself, just as is the case with Freud's definition of the complex, which, according to the greek drama, is articulated by the figure of the parricide. In this sense, Derrida claims that Freud "has shown

how this archontic, that is, paternal and patriarchic, principle only posited itself to repeat itself and returned to re-posit itself only in the parricide" (ibid., p. 60).

**Institutional Passage** What is the case when this patriarchic principle is found on archives? I mentioned above the passage from the private to the public that Derrida recognized in the concept of the archive. This fact resonates with the plurality that is condensed in the place of the archive, which is opposed to the singularity that is condensed in the plurality of memory. In other words, while archives are a singular place where documents —made by a plurality of authors, that is, by different people, technologies, etc— are singularly kept, in the case of human memory, the reverse occurs. While the psyche is singular, that is, while there is only 'myself' remembering, the difference between each trace of memory is indeed a plurality. While I remember, the memory I rememorate is composed out the set of differences of traces that I have effracted in the moment of remembering.

The following pseudocode may help in this distinction:

```
function make_an_archive() {
        public  ← PRIVATE
        singular ← PLURAL
        place  ← COMMUNITY
        archive ← place ← singular ← public
        return archive
        }

function make_a_memory() {
        private ← PUBLIC
        plural  ← SINGULAR
        trace  ← SELF
        memory  ← trace ← plural ← private
        return memory
}
```

**Authorities** For example, the case of making an archive can be exemplified with the passing of a law or the signing of a contract. Derrida speaks of the act of consignation as constitutive of the archontic principle, since it is an act of "gathering together signs" (ibid., p. 10), as is the case of the signatures on a bill or a contract. One's signature is private, therefore, the moment one signs a document one makes it public. Furthermore, in making a contract, multiple signatures need to be gathered into the same space: the plurality of the community needs to be condensed in the singularity of the document. Finally, the document itself comes to be a place, but it itself needs yet another place where it is stored, which comes to be the archive. What this act of consignation entails is, precisely, the archontic principle, since the contract itself now becomes an authoritative presence that emerged out of this function of making an archive. The structure of this presence will be discussed in the following section (See 2.2.4). For now, let us continue with the case of making a memory. The moment one perceives the world, one makes it private, inasmuch as one engages with remembering it. When one begin remembering, oneself begins the process of pathbreaking and effraction of traces —and of forgetting, as I will continue to develop below–, which results in the plurality of the condition of traces, that is, on the difference between traces constitutive of memory.

So, given that if this constitution of archive and memory are indeed reflections of each other, that is, one being the reverse of the other, how is the archontic present in the making of a memory? What would a reversed archontic principle look like? How would it act? These questions deserve a few paragraphs because they will explain the relationship between database, memory, and archive.

**Anarchic Memory** What is important to note here, before continuing, is that just as memory is in a state of fracture and rupture, an archive is also in such state of discontinuity, and thus it is this condition of being in the form of disconnected gaps that makes memories and archives so alike. Spieker notes is that of discontinuity and rupture: "Like all kinds of data banks, [the

archive] 'forms relationships not on the basis of causes and effects, but through networks"' (Ernst, 2013, p. 113). From these two qualities of archives (filtering and fracture), their resemblance to memory can be drawn. However, while the archive is indeed patriarchic, memory, on the other hand, in terms of the archontic principle, it can only be described as an-archic. While there is indeed oneself remembering, there is no possibility of ruling over the plurality of traces that one has in memory. There is no place to go to remember things, one may try to trick the body into feeling similar things as when one was remembering so as to make a memory resurface, but in the end, memories remember themselves. One has no control over neither remembering nor forgetting. Memories forget themselves.

**Collective Memory** In the fantastic case of Funes (See 2.2.1), for example, forgetting is a mysterious lack that cannot be reduced to a simple loss of information. It is also a necessary condition for memory to be considered as trace, given that the moment a trace begins is when its own erasure begins. Thus, every memory comes with its own forgetting mechanism, a mechanism that is triggered on its own, and, further, that it is still not fully understood (Wessel & Moulds, 2008, p. 292). However, in the case of an archive or, what some (perhaps wrongfully) refer to as 'collective' memory, the opposite occurs: data loss consists of plain and simple erasure, that is, a destruction that cannot be considered an instance of forgetting: "in collectivistic memory, where the database has a tangible form, it is more apparent that permanent loss is a possibility. In archives, ink may fade, paper may crumble and entire files may end up in the shredder. (ibid., p. 292)"

**Writing Code** Since writing can function as a link between human and the nonhuman memory (See 2.2.2), several instances of the metaphor of writing need to be explained in relation to databasing. For example, not only programming languages imply writing in terms of symbols (words and characters), also data structures appropriate the concepts of writing and erasing. Furthermore, in resonance to the Derridean trace, erasure is embedded in the structure of writing. This is to say,

that C++ classes include within their own data structure a call to their `destructor`. This means that, whether explicitly or implicitly, all classes —i.e., all data structures which correspond to instantiated objects— have a way to self-erase, or self-destruct after the object is no longer needed. This self-destruction means, precisely, releasing the object's resources, that is, to free the physical memory space that it has occupied throughout its lifetime.[9]

**Anarchic Computer Memory** This comparison between `destructors` and forgetting serves as a starting point to determine the extent to which computer memory —i.e., data structure handling in restricted, discrete space— can be thought of as human memory itself, and, if so, the extent to which it also constitutes an instance of an-archic structure. (The first thing that comes to mind is an an-archic program written in C++ with self-destructing classes that fire at will, in complete unpredictability, rendering the software utterly anarchic, but utterly useless.) Therefore, there has to be a different way to think of an-archy in software. In what follows, I address the extent to which databases can be understood in terms of memory and anarchy, taking German media theorist Wolfgang Ernst's concept of the *anarchoarchive* to a different dimension.

### 2.2.4 The Spectral Database

> The structure of the archive is *spectral*. It is spectral *a priori*: neither present nor absent "in the flesh," neither visible nor invisible, a trace always referring to another whose eyes can never be met... (Derrida & Prenowitz, 1995, p. 54)

**Computer Memory and Writing** Despite the multiplicity of elements that constitute software programming, such as compiler instructions, hardware stipulations, collaboration platforms, etc., the concepts of inscription and erasure (writing) are still in place. Because of this, the concept of the Derridean trace applies to programming. This fact can be linked further back to the conception of the computer itself, as it was previously discussed (See 1.2.1). In his architecture, Von Neumann

[9] https://en.cppreference.com/w/cpp/language/destructor

proposes that the storage unit of the computer allowed data to be written and erased in different locations and times. In fact, he was following Turing's conceptualization of the *a-machine* —i.e., the *Turing machine*—, which was a mathematical model for computation, that can be represented by a symbol scanner and an infinite tape, where the scanner gets, sets, or unsets a symbol on the tape, and the tape moves to the next slot accordingly. Therefore, these setting and unsetting movements represent inscription and erasure, to the point that, as Kittler notes, quoted by Ernst: "the two most important directing signals which link the central processing unit of the computer to external memory are being called `READ` and `WRITE`" (Ernst, 2013, p. 131).

**Memory Replacement** However, an important distinction needs to be made here. While I am arguing for the similarities that exist between memory and database, Ernst proposes that databases —i.e., what he refers to as digital an-archives— enter as a form of replacement. This is to say that, given the convergence of all media into digital media, 'reading' is rendered null, giving way to mathematical processes that come to interpret the data. In Ernst's words, "signal processing replaces *pure* reading" [emphasis added] (ibid., p. 130). This statement is only valid within the Kittlerian, disembodied worldview (See 1.1.3), which would also allow to convey the following: signal processing replaces *listening*. Thus, in such a world where data structures and algorithms perform our own capacities to create and, by doing so, remove our bodies from ourselves, there would be no need for neither reading nor writing. Since, if the database reads itself, it also writes itself, and, in the case of music, if it listens to itself, it also sounds itself, removing listening altogether. Therefore, in order to arrive at the point of resonance that enables both the nonhuman and the human to coexist, a reconceptualization of the anarchic in relation to databases needs to be made; specifically, one that is grounded not on substitution, but on difference, and one that explains how authority enters into the sphere of databases.

**Anarchic Records** In his conceptualization of the *anarchoarchive*, Ernst (ibid.) opposes technical recording with symbolic transcription. Given the fact that a microphone captures the entire sonic environment, this involuntary memory —of "past acoustic, not intended for tradition: a noisy memory" (ibid., p. 174)— comes to be a form of anarchic archive, or *an-archive*. Therefore, for example, he claims that Bela Bartok's transcriptions to musical notation of Milman Parry's Serbian epic song recordings[10] becomes an archivization process, that is, a process by which symbolic transcription leads to an ordered archive, i.e., a score.[11] In Ernst's sense, a sound file storing samples of the recorded world is anarchic because of the unfiltered ('pure') way in which the world is inscribed by the recorder. He bases this idea on another media theorist, Sven Spieker, who claims that, following Derrida's conceptualization of the archive, a central feature of archives is not memory, but a need to "discard, erase, eliminate" that which is not intended for archivization (ibid., p. 113). In other words, the possibility condition of an archive is filtering, or better: framing. The problem is that, the moment Ernst deposits on the recording technology the disappearance of the body, he dislocates framing altogether, and thus is how the concept of 'pure' recording comes in. Therefore, if for Ernst, Parry's audio recordings are anarchic, this is because they are constructed upon a Kittlerian idea of purity that leaves the human out of the equation.

**Memory and Framing** If we consider the body's capacity to create images from the world of images, then framing is also a possibility condition for human memory. In fact, we have seen with the Derridean trace that it is a resisting force what first enables the violent rupture of traces, and thus, this force can also be considered as an instance of framing itself. Resistance of the psyche as creativity taking action as the moment of framing. In the an-archic memory graph that I am conceptualizing here, framing can be thought of as the edge (the arrow) that thus connects the public to the private. Furthermore, framing can also be inversely considered as the arrow between

[10] https://mpc.chs.harvard.edu/

[11] Another example Ernst provides of the anarchive is the Internet itself: " [The Internet] is a collection not just of unforeseen texts but of sound and images as well, an anarchive of sensory data for which no genuine archival culture has been developed so far in the occident" (Ernst, 2013, p. 139).

the private and the public in the case of archives, since, for example, in the case of a contract, only a limited amount of signatures are needed, thus a frame needs to be an active part of the process of archivization.

**Nonhuman Tympans** Therefore, given that framing is in between the public and the private, in order to consider the case of audio recording as either memory or archive, these two categories, privacy and publicness need to be taken into account in relation to nonhumans. In other words, given that the world can be considered public, because it is in a constant state of availability at any time, then, a microphone can be considered an actor of privacy, since it deprives some sounds from being recorded —e.g., because it has not been designed for certain frequency bands— while recording other sounds. Therefore, if transducers have a filtering capacity, they engage with the passage from the public to the private —and vice versa, since both speakers and microphones are transducers. Consider, for example, Jonathan Sterne's definition of the 'tympanic' function of transducers, as quoted by Cathy Van Eyck in her PhD dissertation on microphones and loudspeakers: "every apparatus of sound reproduction has a tympanic function at precisely the point where it turns sound into something else... and when it turns something else into sound" (van Eck, 2013, p. 107). Therefore, considering these nonhuman tympans as actors of privacy and publicness, audio reproduction technology can be compared to the structure of memory and archives. For example, audio recording can be conceived as a form of memory, only to the extent that its inscription is singular yet reproducible. This is because a magnetic tape or a hard drive cannot be considered in the same (plural) way as Derridean traces, since once data is stored, it exists only once in discrete, limited space, unless copied to another location, in which case it becomes a duplicate, an identical (in-different) clone of itself. Furthermore, the case of audio playback can be a form of archive, since it consists of the passage from the private —the stored air pressure waves— to the public —the reproduced air pressure waves in space—, only to the extent that the latter is not considered space as such, but resonance within space. In this sense, Ernst' consideration of the

Parry's recordings as anarchic needs to be reconfigured.

**Spectrality of Archives** Given that these nonhuman tympans constitute the limit between sonic —but also tactile (ibid., p. 223)— world and the binary world of databases, their comparison to memory and archives renders a hybrid object: one that, on the one side, becomes a private and singular 'trace', and, on the other, becomes a public, resonant 'space.' Thus, databases represent neither a trace nor a space. At this point it is important to address the quote at the beginning of this section, that is, the structure of the archontic that Derrida assigned to archives: spectrality. Derrida claims that, addressing a phantom is a "transaction of signs and values, but also of some familial domesticity" (Derrida & Prenowitz, 1995, p. 55), meaning, on the one hand, that in the uncanny encounter with a ghost, there is familiarity, that is, there emerge feelings of what is known to be close to us, but also that which composes the authority of that closeness. Therefore, embedded in this familiarity is the archontic, the oedipal, etc., and thus the expression of power that this apparition brings forth. On the other hand, the familiar is also related to an economy —Derrida points to the greek root *oikos*, meaning 'house'—, that is, to the passing through (trans-action) of signs, but also de translation —or better, the 'transduction' of things. This is why Derrida considers any encounter with the spectral to be an instance of addressing, that is of transaction. Finally, what this uncanniness of the ghost entails is nothing other than the haunting itself, as Derrida writes: "haunting implies places, a habitation, and always a haunted house" (ibid., p. 55). To bring back our database, the hybridity that the database projects when compared to memory or archives points to a certain uncanniness, that is, precisely to the hauntedness that comes from its spectrality. Furthermore, it can be argued that considering the database in such way is plain and simple delusion, that is, insanity at its best. Not surprisingly, this is the point exactly; within this delusion exists truth:

> . . . it resists and *returns*, as such, as the spectral truth of delusion or of hauntedness. It *returns*, it belongs, it comes down to spectral truth. Delusion or insanity, hauntedness

> is not only haunted by this or that ghost. . . but by the specter of the truth which has been thus repressed. The truth is spectral, and this is its part of truth which is irreducible by explanation. . . (ibid., pp. 54–56)

**Spectrality of Databases** Therefore, given that Derrida considers the structure of the archive to be indeed spectral, I can bring this spectrality to the database itself, and consider it just as spectral but in a different sense. The spectrality of the database comes not from its hauntedness, that is, it is not domiciled since, as humans, we have no access to data space (address space) directly with our bodies. As humans, we cannot engage in transaction with the specter directly: we need transducers. In the case of audio playback, we need loudspeakers because we need to create, with our bodies, the image that was recorded in the first place. In consequence, while embodying stored data, we embody its specter. Therefore, the hauntedness needs to be 'transduced' into space, and with this transduction the agency of the database can be effectively felt. Likewise, the uncanniness of the case of audio recording can be felt as the ghost that will 'remember us,' because of this computer 'memory' that will keep (to its own privacy) the sonic environment that attests to our presence in space —because we sing, we make sounds, etc. Hence, the haunting that resides in-memory, together with the stored data and pointers.

**Agency of the Uncanny** In recognizing this presence, which is the archontic specter of the database, comes the recognition of the agency of the database itself, and, furthermore, of the quality of the aesthetic experience that we encounter whenever there is a database in art. As Timothy Morton (Morton, 2013) writes in relation to what he calls hyperobjects:

> Recognition of the uncanny nonhuman must by definition first consist of a terrifying glimpse of ghosts, a glimpse that makes one's physicality resonate (suggesting the Latin *horreo*, I bristle): as Adorno says, the primordial aesthetic experience is goose bumps. Yet this is precisely the aesthetic experience of the hyperobject, which can only be detected as a ghostly spectrality that comes in and out of phase with normalized human spacetime. (ibid., p. 169)

To the point that databases are spectral, they can be considered hyperobjects in Morton's

sense, and thus, agents translating through aesthetic networks. However, I will not dwell much longer on this definition in this text. If we consider the "recorded movement of a thing" with which Latour identified his concept of the network, databases are agents not only of recording, also of motility, and of thing-hood itself. This is to say that, in the constitution of networks, the spectrality of the database expands in multiple directions. Bringing back the illusory violin of the acousmatic concert I mentioned earlier (See 2.1.2), it exemplified Hansen's concept of the creation of images our brain's capacity for virtuality has, that is, our imagination. The sound of a violin can be recorded —but also synthesized— into the privacy of a database. Then, it can be played back with loudspeakers located in such a way that they emulate the location of an actual violin player. As a result, the listener could very likely imagine a physically present violin in the room, that is, a ghost. This ghost comes in as the phantom of a human player; of the violin itself; of the histories and traditions that those two elements bring forth; of the presence of the nonhuman that the database implies; of the privacy that is not human but that it is still uncannily private; of the plurality that is embedded in the construction of databases; of the hauntedness of the archontic that the above sets forth; and so on. In this way, the spectrality of the database attests to its relation to memory and archives, and, thus, to its aesthetic resonance within our experience.

In what follows, a crucial aspect of the database will be addressed, one that defines the condition of possibility of this hauntedness to be indeed instantiated into appearance, not only in the form of authority, as I have shown in the case of its archontic presence, but also in the form of style, and most important, gender: the performativity of the database.

## 2.3 Performativity Of Databases

### 2.3.1 Gendered Database

> Gender is not passively scripted on the body, and neither is it determined by nature, language, the symbolic, or the overwhelming history of patriarchy. Gender is what is

> put on, invariably, under constraint, daily and incessantly, with anxiety and pleasure, but if this continuous act is mistaken for a natural or linguistic given, power is relinquished to expand the cultural field bodily through subversive performances of various kinds. (Butler, 1988, p. 531)

Judith Butler (ibid.) distinguishes between an expressive and performative self. The former comes from an essentialist view from the self as being 'inside' and displaying itself on the outside. The latter is an illusory self, strictly outside and unrelated to the "natural or linguistic given." She understands gender within this performativity of the self. Like the self, gender emerges temporally, at the surface level of the skin of the body. This notion of gender relates with Jean-Luc Nancy's notion of resonance and the self (See 2.1).

**Skin of the Database** In the performativity of databasing resides the possibility for the skin of the database to emerge. On the one hand, this skin is the spectral surface of the database's illusory self. On the other, it is the limit towards which the human and the nonhuman engage in resonance. The skin of the database carries the mark of a style. That is to say, in defining style as a repetition of acts, it is a form of embodiment that is ascribed to databases. Because the self of the database is inscribed on its skin, we can assign yet another quality of the database: an authorial subject. This subject relates to the uncanny, not only because of its nonhuman spectrality, also because of its gendered appearance which comes to redefine our own social categories such as reality itself. Therefore, I claim that, given that databasing is the performative condition of databases, the database becomes gendered.

**Expressing Nothing** Butler sets forth a critical genealogy of gender which relies on a "phenomenological set of presuppositions, most important among them the expanded conception of an 'act' which is both socially shared and historically constituted, and which is performative…" (ibid., p. 530). Furthermore, the difficulty Butler recognizes in this view of gender, is that "we need to think a world in which acts, gestures, the visual body, the clothed body, the various physi-

cal attributes usually associated with gender, *express nothing*" (ibid., p. 530). Therefore, how can the database itself be conceived in this terms of performativity, to the point that, while having the capacity to store millions of data, a database can in deed express nothing?

**A Historical Situation** Butler defined gender identity as a historical situation, distinguishing between physiological facticity of the body (sex) and the cultural significance of such facticity in terms of gender. Added to this distance between body and identity, Butler speaks of the body as performative process of embodying cultural and historical possibilities. These possibilities, which are delimited by historical conventions, are thus materialized on the body: "one does one's body differently from one's contemporaries and from one's embodied predecessors and successors" (ibid., p. 521). Therefore, the body comes to be a "historical situation" that results from the performativity of embodiment itself. In other words, the actions related to what Butler calls the "structures of embodiment" constitute an ontological sphere of present participles, such as 'doing,' 'dramatizing,' and 'reproducing.' Furthermore, what this structure of embodiment entails is the constitution of not only gender, but also style. Since gender is constituted temporally, it is necessarily historical:

> to be a woman is to have *become* a woman, to compel the body to conform to an historical idea of 'woman,' to induce the body to become a cultural sign, to materialize oneself in obedience to an historically delimited possibility, and to do this as a sustained and repeated corporeal project. (ibid., p. 521)

**Subversive Repetition** Far from being a prescribed given, the constitution of gender on the body is itself a result of mediated history. In other words, gender is a creative act of interpretation and reinterpretation that reveals itself on the body, not as an expression that comes from within, but as the sedimented layerers that deposit themselves in time. Furthermore, within this notion of temporality there is the need for repetition, but a repetition that is susceptible to breakage, or what Butler refers to as "subversive repetition" (ibid., p. 520). In being a temporal identity which reveals

itself through a "stylized repetition of acts" gender constitutes an "illusion" of a gendered self. These acts take place *on* the body, by the mundane instantiation of bodily gestures, movements, and enactments. Furthermore, the these acts are necessarily discontinuous, and it is because of this discontinuity that exists the possibility of gender transformation. In this sense, gender performance is neither linear nor nonlinear. It resides along an anarachic temporality that replaces teleology with the multiplicity of resonant nows. It is an inline iterative function with random breaks.

**Gendered Database** The database is a collection of facts. It is facticity itself. This is what Butler's gendered self can teach about databases: in performing the database, the database appears like gender, as a historical situation. Its body is felt neither as the database body, as if the materiality of the computer's architecture could come as a proxy for the nonhuman body; nor as the extension of the embodying databaser, that is, as a prosthesis that expands the databaser in an expressive way. The body of the database emerges as a phantom, as spectrality itself, and it is this nonhuman presence that engages in the publicness of performative acts. The specter of the database must not be understood spiritually, or as a *deus ex machina*, or as a soul, or singularity that begins to act *as* human and, by extension, supersedes human. It is simply a nonhuman fabrication of selfhood: there, around, making its way through the rupture of the permanent condition of performativity to which we (humans and nonhumans) are phenomenologically bound. That is the how the style of the database appears. This nonhuman self, like Butler's gendered self, is equally 'outside,' that is, "constituted in social discourse" (ibid., p. 528). That is to say, the skin of the database is open for perception outside of itself, and in fact, nothing of the database can be considered expressive. Inside the database there is literally nothing but zeros and ones, nothing but data; in the same way, nothing is inside of the body but flesh, bones, and veins. When considered as internal, inherent, or essential, the classical notion of the self, in its heteronormativity, is seen as a "publically regulated and sanctioned form of essence fabrication" (ibid., p. 528). In this state of being fabricated, expressivity serves as the foundation for what Butler refers to as the 'punitive'

aspects of wrong gender performance. In this sense, the social quality of acts that fall outside the regulated binary gender construction work their way into punishing the body, incarcerating it, severing it, as is, for example the famous case of Turing himself:

> Turing's later embroilment with the police and court system over the question of his homosexuality played out, in a different key, the assumptions embodied in the Turing test. His conviction and the court-ordered hormone treatments for his homosexuality tragically demonstrated the importance of *doing* over *saying* in the coercive order of a homophobic society with the power to enforce its will upon the bodies of its citizens. (Hayles, 1999, p. xii)

### 2.3.2 Towards The Limits

**Exposure** The performativity of databasing can be understood in terms of what Nancy calls exposure (Nancy, 1991). Exposure is the appearance of a limit and the finitud of a singularity. With this limit instantiated in the (public) moment of the performative act, is how communication emerges as that which is in common among singularities. That is to say, because it is itself nothing ("neither a ground, nor an essence, nor a substance" (ibid., p. 31)) Nancy considers finitud to just appear in the form of communication: "it presents itself, it exposes itself, and thus it exists as communication" (ibid., p. 31). His emphasis on communication as exposure marks a crucial distinction on the concept of community. For Nancy, as I have described above (See 2.1.4), community cannot come from an instance of work: it emerges as an instance of the communicative action which is "the unworking of a work that is social, economic, technical, and institutional" (ibid., p. 31). In other words, the performativity of this communicative act, the publicness and dramatic qualities with which it unfolds, result in exposure. This is why, in the case of databasing, what is exposed at the limit of its performance is the finitud of the database itself. Through this exposition of the limit is how the singularity of the database can be communicated; or, better, through this exposure is how communication expands our concept of community towards one that includes the database as an agent of community itself.

**Anarchic Touch** As Nancy writes, "a singular being does not emerge or rise up against the background of a chaotic, undifferentiated identity of beings" ibid., p. 28. This is to say that, like Butler's gendered self, there is no substance within singularity. The appearance of a database as finitud comes not from an originary *archē* which would impose its archontic power as is the case of archives. Further, the limit of the database is not instantiated out of the one, a "unitary assumption" ibid., p. 28, or the wholeness of a single one. This finitud does not come from intentionality or any essentialist notions: it simply appears, for Nancy, in the form of touch: "... as finitude itself: at the end (or at the beginning), with the contact of the *skin* (or the heart) of another singular being, at the confines of the same singularity that is, as such, always other, always shared, always exposed..." [emphasis added] ibid., p. 28. The temporality of this touch, like that of gender, is anarchic.

**Communities of Skin** The limit of the database, as performative, spectral skin, is the condition for community to emerges between the human and the nonhuman. This means that the agency locus of the database needs to be placed precisely on its skin, because it is what becomes public of itself. In other words, given that this skin is available to the perception of others, it becomes touchable, it reaches our own limit as databasers. By doing so, that is, by exposing our own limits to ourselves and to each other, the database changes our definition, or, better, delimits the extent of our own singularity. However, this does not mean that it stands in the way of our performativity, or worse, that it precludes or determines ourselves. If this were true, we would be once again subject to technical determinism, essence fabrication, etc., and falling out of considering any possibility of community between anything that is not human —or, more accurately, anything that is not 'man.' As Nancy rightfully claims, " it is not obvious that the community of singularities is limited to 'man"' ibid., p. 28. Thus, the fact that the skin of the database changes our own skin simply means that we are already in communication with it, that is, in community, and also in a state of resonance with it. This is the function of the skin of the database: like the skin of a drum, or the skin of a loudspeaker, the skin of the database resonates with our own skin, engaging the resonant body

with a resonant spectrality. This is the community of resonance, the community of the resonant network, which has no purpose, no intentionality behind, no essence; only appearance and motility, performance and repetition: an activity that is the unworking with which community exposes itself.

**Hybrid Pluralities** Database models tend to reside next to each other, either within a single database system or within an interconnected networked system. With this plurality of the model, databasers have access to the many features that each model offers, focusing on those features that are suitable for their needs. The skin of the database is as fluid as the constitution of gender, and if this is true, then the fluidity of databasing itself comes to represent the constitution of gender through the performativity of databasers. By resonanting in such performativity, databasers approach (but do not reach) the limit of the database. This approach to the skin of the database exposes simultaneously the skin of the databaser to the database. What this exposure amounts to is not, however, an opposition of forces. It results in the fragmented state of community that resides in the different degrees of this exposure. In other words, this exposure is of a hybrid plurality that resonates at the limit. Engaging with the touch of the spectral database means reconfiguring, resounding, and remembering our own sense of touch, just as well as our own sense of self.

### 2.3.3 Contingencies Of Style

> . . . style, supplementing timbre, tends to repeat the event of pure presence, the singularity of the source present in what it produces, supposing again that the unity of a timbre —immediately it is identifiable— ever has the purity of an event. . . The timbre of my voice, the style of my writing are that which for (a) me never will have been present. I neither hear nor recognize the timbre of my voice. If my style marks itself, it is only on a surface which remains invisible and illegible for me. (Derrida, 1982, p. 296)

A database without performance represents a disembodied ‘base’, that is, the spatially ordered set of computer hardware together with the software routines that it embeds. It is its most basic level, a foundation upon which the database tree can be performed. This ‘base’ in database

comes as a stage for databasing itself: a stage without performance is an empty stage, extension of space itself. Databasing projects its own style as a result of its performance, and through this projection comes the exposure of its skin. The "stylistic repetition of acts" in the dramatic case of the gendered database is now revealed as style itself. Like skin and voice, singularity emerges as style and timbre.

**Style and Timbre** 'Style' comes from the latin *stilus*, meaning a sharp object with which you can write: like the stylus of a record player, it is a writing tool. Its meaning extends through writing to the manner in which 'writing is carried out: the variations and oscillations of the pen and of the text itself, hence resulting in the style of a certain text, or, for that matter, a programming style, or even the style of an author. Beyond writing, style becomes the way in which the body moves, how it looks, whether it is human or nonhuman: the style of a music work, the style of a composer; and beyond, the style of an entire musical period, thus extending style in time and space. Most important, style is a manifestation of the singular. In the sense that style does not lend itself to duplication, and provided that it happens as the apparition of an event, it exposes singularity as such. Style is thus comparable to the voice of a certain author, and also to the sound of the voice itself: timbre. That is to say, style and timbre can be understood equally as the presence of the singular: the signature that comes with the unique and irreproducible timbral quality:[12]

> In its irreplaceable quality, the timbre of the voice marks the event of language. By virtue of this fact, timbre has greater import than the *forms* of signs and the *content* of meaning. In any event, timbre cannot be summarized by form and content, since at the very least they share the capacity to be repeated, to be imitated in their identity as objects, that is, in their ideality. (ibid., p. 296)

**Endless Databases** The skin of the database unfolds in the duration of the performative act. What is exposed as its singularity is the ruggedness of the traces of which it is composed. That is

[12] In signal processing terms, the sound of a voice might be approached with timbre stamps or vocoders, a type of Fourier-based filter in which "the spectrum of one sound is used to derive a filter for another" (M. Puckette, 2007).

to say, the discontinuities of its reticulated constitution of style. The length of this skin can only be estimated: there is no possibility of rendering it complete. In this fractured state it points to infinity. In this sense, databasing means participating in the infinite, taking a small part of the infinite: performing the infinite within the limits of our own embodiment. Furthermore, the contingent situation of resonance within the frayed spatio-temporal configuration of networks relates to the concept of chaos. I have mentioned earlier the relation between computers and users as understood in terms of complex systems (See 1.2.2). Considering databasing as chaotic systems brings yet another aspect to the contingency of style.

**Database and Chaos** Given that this style can be considered as an emergent singularity of databasing, this singularity can be considered as well deterministic. In mathematics, determinism refers to the capacity to predict results, specifically by solving differential equations. This is the case of dynamic systems studied within Chaos Theory. For example, the Lorenz attractor is a system of differential equations discovered by Edward N. Lorenz in 1963, following experiments on weather conditions prediction. The attractor is most famously recognized by the butterfly-like appearance of its visualization, which is also related to the concept of the 'butterfly effect' (See Figure 2.3). The Lorenz attractor is a dynamic system, which means that it can render very different and quite unpredictable results by minimal changes on their initial conditions, despite the fact that it is indeed a deterministic system. Furthermore, a graph of a dynamic systems presents fractal properties. Considering, thus, databasing as a dynamic system, on the one hand, two performances can be exactly the same if given the same initial conditions and states. This is the case, for example, of the performance of fixed media (digital) works, which at least at the sample level, every bit of it is exactly the same of the original.

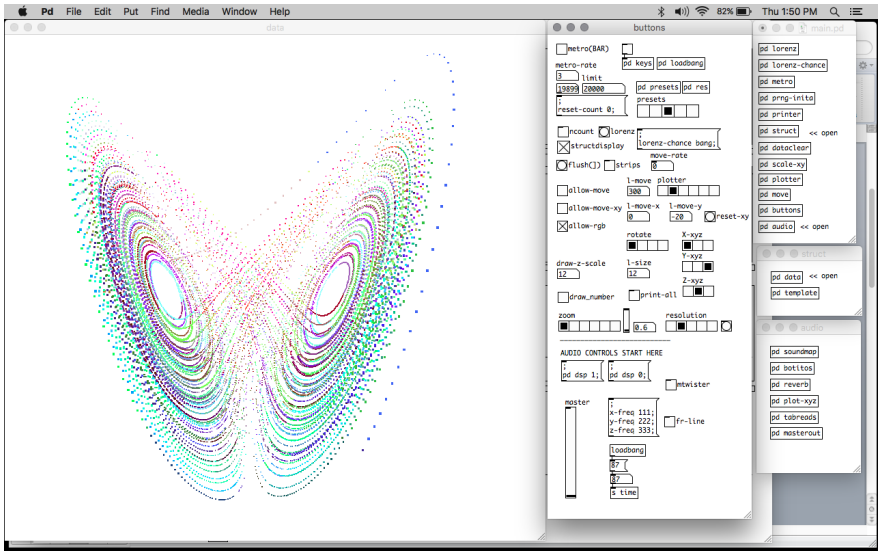



Figure 2.3: Lorenz Attractor
Plotting of the Lorenz system in Pure Data.

**Fractality** However, identifying predictability in this way means falling in a cybernetic trap, of which Hayles already warned about when considering Turing's Test. Hayles reads Turing's test as a game which, in order to play you are already part of its outcome because you accept its predicates as a condition for playing. In Turing's case, the moment you enter into the disembodied place where the screen is the only thing you see, you are already a cyborg, and the definition of the human and the nonhuman is already laid out in principle. On the one hand, by equating fidelity of data storage with fidelity of performance, one is already removing the human out of the concert stage, and the question of performance altogether, leaving only the idealist and romantic notion of the work of art in its pure and objective state. On the other, in order to allow for the style of databasing (skin) to emerge, one has to consider not only the actual staging of performance, also the staging of listening itself, which is the possibility condition for the resonant subject of the database to emerge as the communicative apparition of a skin. Therefore, the contingency of style (as chaotic state) can only emerge out of the unpredictable agency of the unfolding. This is how I consider databasing and the contingency of style: the unpredictability of databasing has the

qualities of a fractal. On the one hand, because of the fractal dimension, it expands the definition of geometric figures to the infinite. On the other, it presents an unfolding symmetry (self-similarity), which relates to their shapes being replicated nearly exactly in different scales.

**A Music Work as a Singularity** This aesthetic experience, it must be noted, is of a nature that slips through the cracks of traditional conceptualizations of the work of music as a result of stylistic, or stipulated constraints on the part of the composer, or stochastic procedures. For example, composer Horacio Vaggione goes to great lengths to prove that the musical work affirms itself as singularity, in the particular sense that its rules are only prescribed from within, and always in an "action-perception loop" with the composer (Vaggione, 2001). What Vaggione is arguing against, is the tendency of formalized musical processes that had reshaped the black-boxed approach towards composition in CAC: "a composer [unlike a scientist] knows how to generate singular events, and how to articulate them in bigger and bigger chunks without losing the control of the singularities" (Vaggione, 1993, p. 97). However, there is a fundamental concern that needs to be addressed in relation to the contingency of style. For Vaggione, style comes to represent the reified status of the rules within a work, insofar as this reification is taken for the starting point from which to compose, and not the result of a composed thing. Consider this quote from an earlier text:

> Here lies what seems to be one of the sources of confusion regarding the nature of music composition processes: on the one hand, we must make as careful a distinction as possible between the collective rules and the composer's own constraints; on the other, this distinction seems irrelevant [because] any primitive (coming from a common practice or postulated ad hoc) is to be considered as a part of what is to be composed, *in order to produce a musical work affirming itself as a singularity, beyond an exercise in style*. Adorno was of course conscious of this dialectic: his statement about sound material considered not as something "given" but as a "result" of a musical thesis clearly points to this fact. [emphasis added] (Vaggione, 2001, p. 59)

**Arbitrariness** Despite the fact that, distinguishing between rules and constraints, that is, between socially and historically established canons —as stylistic conventions—, and locally established

postulates to be carried out by the composer —as constraints—, is crucial in defining style in databasing, at the same time both ends collapse into the realm of arbitrariness that composition is made of, for, if any "primitive" of the composition is indeed to be considered "part of what is to be composed," then style itself becomes a result. This is what Vaggione means by "beyond an *exercise* in style:" it is not an exercise in the sense of a draft, in the military context of training (practice for the sake of training). Style is not an exercise because it cannot be operative in the sense that it is considered a product of work. That is to say, given that style is the contingent skin that emerges out of the performative action of databasing, if it were considered productively, that is, in terms of a targeted object towards which work is oriented, then it would only result in closure, in a closed object, and thus, it cannot be considered contingent, precisely because it is stipulated from the start as a law to which every composable element abides. This is the case of, in Vaggione's sense, formalized processes which act as global laws, and in which —comparing it with a marching army following the one-two directive: "no singularities —and hence no specific formal (relational) properties— are here allowed" (Vaggione, 1993, p. 101).

**Inoperative Style** However, if style is understood as inoperative, that is, as the unfolding of the performative act of composition, which implies equally the 'composability' of all primitives, then, its contingency appears in the form of exposure, not as a closed object, but as an unclosed object, some *thing* that is exposed and bound to exposure; a thing that exposes us in the same resonance of its touch. Like the marks on our skin, like its wounds; like the cracks of an old house, like debris, wreckages, or any form of residual mark that is the evidence of an event; with forensic intimacy, the contingent style of a musical unwork reveals itself as communication. This is what connects aesthetic experience of style with forensic (musical) analysis as well as with an encounter with the spectral. Furthermore, this is how the spectral itself cannot be but a result of the inoperative, of that which escapes the limits of the work, and that which, by releasing itself from itself, returns to itself like the timbre of the voice. This voice of the unwork is what is 'invisible' to it, or better, it

is what it can never listen, and in being hidden or silenced from itself is how it becomes available for listening, what begins listening at the first staging of the waves.

### 2.3.4 A Specter Of Authority

> Gender is instituted through the stylization of the body and, hence, must be understood as the mundane way in which bodily gestures, movements, and enactments of various kinds *constitute the illusion of an abiding gendered self*. [emphasis added] (Butler, 1988, p. 519)

The figure of the author (composer/databaser) is, to a certain extent, exploded as network by the complexity of the system. However, authority is reified into the name because of the interplay among work, productivity, and product. In this section I attempt an approach to the name of the composer not by its work, but from the illusory perspective of authority. However composable all Vaggonian primitives can be, the structure of the database tree is so vast that any attempt to comprehend it as a whole would extend it even further (See 2.1.3). However, this determines neither the extent of the performativity of databasing, nor the agency of the human. Quite the contrary, expansion through the network can be considered as the trace of the author, or better, the elongation of the spectral shape of an author. Further, with the performativity of databasing, the databaser too becomes incomplete.

**The Name** The infinitude in the fractality of databasing, however, is at some point reified in a figure or a name. This figure is the place where authority is condensed, and it responds to traditionally essentialist conceptualizations of the romantic author which, despite the many attempts during 20th century,[13] are still in effect today, specifically in the field of music composition. It is not the purpose of this section to criticize this tradition, namely because I don't consider it relevant for the purposes of databasing. Focusing on it would be missing the point. That is to say, in the case of

[13] See for example Roland Barthe's 1967 *Death of the Author*, or Michel Foucault's 1969 text *What is an author?*, both of them commented on in (Daniel, 2007).

databasing, such figure of an essential author is simply dislocated and forced upon the structure of the network, and it is anachronic because it constitutes a temporality set against the temporality of networks. Databasing, as resonant performativity already exists beyond this traditional figure of the author. However, in its spectrality that stems from the archontic (See 2.2.3), authority can be seen as the illusory resonance of an author. It is this illusion that I attempt to address here, this ghost which haunts music composition.

**Dictionaries** Consider how style is used in some cases of CAC. David Cope's Experiments in Music Intelligence (EMI) (Cope, 1987b), for example, can be considered a formalization of compositional authority. That is to say, intentional stylization "based on a large database of style descriptions, or rules, of different compositional strategies" (IV, 1999, p. 3). Written in the functional programming language LISP, EMI's focus is "style imitation" in order to assist the composer when in front of a "composing block," provoking the "author into almost immediate action. Any blank moments along the way are immediately filled by simple queries…" (Cope, 1987a, p. 38). Cope's approach is inherently hirearchical, and thus based on the premise that music is a language. Therefore, Cope designed dictionaries (databases) of MIDI scores representing the internal relations between composed elements. From items in the dictionary, logically correct inferrences are drawn (predicate calculus) (Cope, 1987b, p. 1). Thus, EMI is aimed at generalizations that reify the authority of the composer as style:

> Years of consistent interactive use have resulted in dictionaries which so complement the author's own style that compositions show little evidence of the origins (man/machine) of the music. (ibid., p. 179)

**Artistry** Vaggione, in response to a formalized approach to music —among many that exist in the literature (Ariza, 2005a; Hiller & Isaacson, 1959; Truax, 1976; Xenakis, 1992)—, proposes the equal role of the informal (or craftsmanship) of the composer using computers. In a very different case of the use of databases, consider Roads' account of Vaggione's workflow when composing

the work *SHALL*:

> These involved arranging microsounds[14] using a sound mixing program with a graphical time-line interface. He would load a *catalog of pre-edited microsound* into the program's library then select items and paste them onto a track at specific points on the timeline running from left to right across the screen. By pasting a single particle multiple times, it became a sound entity of a higher temporal order. Each paste operation was like a stroke of a brush in a painting, adding a touch more color over the blank space of the canvas. In this case, *the collection of microsounds in the library can be thought of as a palette*. Since the program allowed the user to zoom in or out in time, the composer could paste and edit on different time scales. The program offered multiple simultaneous tracks on which to paste, permitting a rich interplay of microevents. [emphasis added] (Curtis Roads, 2001, pp. 313–314)

While this workflow is only representative of certain aspect of the piece in question, it does serve as an example of his concept of craftsmanship. Craftsmanship refers to the manual and direct action of the hand of the composer. The hand, as Makis Solomos very well points out, is not to be understood as being without the tool (mouse) that it needs to use in order to precisely locate sounds on the timeline interface (Solomos, 2005, p. 4). Craftsmanship might be better understood, however, as 'artistry,' thus keeping its relation to hand-made crafts, while maintaining a link with articulation, one of Vaggione's crucial concepts. While articulation relates to the composer's operativity on multiple time scales, artistry relates to the arbitrariness of choice. It is thus a reaction to the abundandy of radical formalism and automation in CAC (ibid., p. 3). Therefore, Vaggione writes, "to write music 'manually', note by note, partial by partial, or grain by grain, is an approach proper to a composer, and he should not be embarrassed about using this aspect of his craftsmanship"' (ibid., p. 3). Vaggione built his terminology not in opposition, but in the spirit of reconfiguring CAC from an embodied stance coming from outside information theory. This stance is not only evident in Vaggione's writings and music. To a debatable extent, it is a point of departure to think of a branch of Argentinian electroacoustic identity that developed in France.[15]

[14]The word 'microsound' refers to sonic events shaped below the threshold of the 'note.' See (Curtis Roads, 2001)

[15]For example, in the work of Beatriz Ferreyra, Elsa Justel, Mario Mary, to name a few. For an approach to Justel's

**The Work of Mice** Instead of being on the rule-based programming of formalization processes alone (keyboard-based input), the artistry of the composer resides in the use of the mouse. The timeline (sequence interface) workflow depends on the pointer. If the presence of the hand of the composer is evidenced by the trajectory (the course of the cursor), it shapes together with the historial of clicks, drag-n-drop motions, etc., the spectral presence of the author. This mouse, as the point of the pointer, the writing device, the 'stilus', becomes that with which we resonate as listeners. Therefore, we perceive the marks of an authorial skin in the database music of pointers. The Vaggionian singularity-based approach to authority embeds composers and computers in a complex system, allowing for the world of music with computers to be a hybrid one. This is how the specter of the author coexists with the specter of the database, and thus, how databasing and composition reveal themselves to be instances of a performativity that resonates aesthetically through the work of music.

## 2.4 Rethinking Composition

### 2.4.1 Interlude: Hyperbolic Reactions

**Imagining Composers** In today's composition and databasing practices, the probabilities of a composer or a databaser working without computers are very slim. Databasing or composition outside the digital seems rather fictional. However, the very image of a 'composer,' which traditionally stems from romantic standards, is already outside the world of computers. This image of composing can be painted as follows: the composer at work, quietly on a desk with pen and paper, transcribing, arranging, making parts, drawing line after line, dot after dot, notating instructions for the performance of an imagined music. Where is the computer in this image of composition? Certainly, placing a computer on this idyllic desk would be anachronic and obtrusive; anachronic,

timeline-based spatialization techniques, see (Cámara Halac, 2018b).

since the romantic quality of the scene would point to the fact that personal desktop computers were not available until late in the 20th century; obtrusive, in the sense that it would attempt against this composer, by interfering with the 'ethereal' link between imagination and notation. This reification of the composer already precludes not only the digital, also the many technological devices that have entered music composition over the years, such as tape recorders, or electronics in general. These technological devices have redefined the composer in many ways.

**Composers and Technology** Georgina Born's ethnography of IRCAM (Born, 1995), captured how the institutionalization of music composition and technology resulted in hierarchical structures of work dynamics, and how these were coated with false notions of collaboration. Inequalities of social, economical, and political status among technicians and composers within IRCAM became privately evident. Knowing how to use computers and knowing how to compose comprised two irreconcilable poles in the institutional structure. For example, Born describes internal hierarchies such as 'superuser' password knowledge, source code access, software licences, and, in some cases, she showed how these hierarchies reflected on internal privacy issues: "workers concocted their various informal ways of protecting privacy and retaining secrecy: blocking the glass walls of their studies, working at night to prevent others knowing what they were doing or even whether they were working at all" (ibid., p. 272).

**Playing with Shadows** On the one hand, it is tempting to link this irreconciliation to the extreme reification of the name Pierre Boulez. The obscure dynamics behind this reification, however privately and secretly they were kept within the institution, can be nonetheless seen as the shadow of the more general specter of the music maker. Born's mysterious but telling anonymization of anyone but Boulez on her transcriptions might attest to this shadow. The music maker has been traditionally considered an outsider, marginalized by society, but simultaneously an integrator of society itself (Attali, 2009, p. 12). On the other hand, this shadow might also be that of the com-

puter itself, the structural presence of a fictional intelligence constructed upon first wave cybernetics. That is to say, precisely because the computer projects an insurmountable power that comes from its calculations, the human is inevitably bound to be subordinate, and with this subordination comes the subordination of the composer, and (perhaps) the end of music. This hyperbolic reaction would explain the need for privacy and secrecy of information, the undocumented "oral culture" of Born's IRCAM, as well as the reversal of the human-computer subordination evidenced in the social strata of the institution. In any case, a composer without computers cannot be imagined today, but this is not due to the practice of composition itself. My argument here is that in any given situation, it is hardly possible to imagine a human without computers at all. This is what media studies has to teach us about the posthuman condition in which we hybridly live, where humans and technology, humans and nonhumans, unfold as interminably networked traces.

**Composers Without Computers** Composing with or without computers cannot be seen as poles on a continuum upon which the name of the composer writes and rewrites itself. Composing cannot be separated from computers, because the human cannot be separated from the non. At the risk of drawing a straw-man out of this computer-less composer, it is very unlikely in today's world to imagine a composer that has not googled 'clarinet multiphonics' for more than a few YouTube tutorials on the topic. The same can be said for digitized music listening, which, in order to escape it, one has to go to great cult-like lengths to do so: going to instrumental performances, getting a vinyl record or a tape player, etc. To have a concert, therefore, a composer without computers today would need to whisper the score to the performers who would, in turn, play by ear. ('By ear', in the sense that they would need to play from memory, since no printed score would exist, for even if the composer wrote the parts, the score would have to be inscribed on a paper, and somewhere along paper networks there is at least one computer.) The composer should also whisper invitations to a few neighbors to be part of the audience. The composer should also demand no recordings whatsoever, while performing for an audience that has been kindly reminded

not to bring their cellphones. Even then, the concert would need to take place on an amphitheater to avoid architectural networks, and Automatic Computer Assisted Design (AutoCAD); before the sun sets, to avoid electricity networks altogether while we are at it; away from cities, a car driving by would be unforgivable; so far away that we would, in fact, need to bring non-perishables for the pilgrimage, and even then, packaging networks or agriculture networks would be almost impossible to avoid. And this is precisely the point: in attempting to avoid it, the pilgrimage exists not in space, but in time, and thus it enters into the realm of fiction. The same can be applied to the overloaded case of a composer totally *with* computers, that is, a computer composer, that would not need the human to write music.

**Databasing Without Computers** The same applies to databasing itself. Removing computers altogether from databasing takes us to the world of libraries, encyclopedias, collectors, gatherers. Most important, it takes us to the place databasing occupies within society: to museums, but also to the dynamics of civilization, to church, put simply: to institutionalization itself. That is to say, it relates performance with the archontic, with the oedipal drive to re-place (See 2.2.3), to an infinite return that the structure of the archive imposes upon us. So, if we imagine a computer-less census, we'd have to picture a gatherer of names walking around town, asking out loud for each person's name and place of residence. Getting rid of networks which might have computers —as in the case of the above painted computer-less composer—, it is clear that the only suitable person for the job would be Irineo Funes (See 2.2.1), and the only possible storage medium would be his memory. Hopefully, the reader would consider this resort to hyperbolic fictions less as a means of justification of the hybrid condition of composition and databasing, and more as an absurd parenthesis that brings nothing of criticism to the —still valid— efforts of working 'outside' the digital. These efforts are not questioned in regards of their validity, only in terms of their definition, which, for the purposes of this text, is understood as built upon organicist notions of the human: the human as the one, indivisible, complete whole. These notions are precisely those that help

reify the image of the composer in its essentiality.

### 2.4.2 Working Composition

> And the listening in question here is not that of a given listener, or of a category of listeners one has to take into account; it is rather structural listening in Adorno's sense —or even, beyond Adorno, a *listening without listener in which the work listens to itself*. ...we hear an organicist concept of the work being strongly articulated [by Schoenberg] (where the work is a whole that doesn't allow any cuts) and a regime of listening whose ultimate, ideal aim is the absorption or resorption of the listener in the work. A listener who is somewhat distracted, inattentive, who would skip over a few tracks daydreaming —*such a listener could fall away like a dead limb*. *Useless*. Bringing nothing to the great corpus of the work. This organicism, in the radical (or structural) tendency that Schoenberg gives it, forms the cornerstone of the construction of a modernist regime of listening. [emphasis added] (Szendy, 2008, p. 127)

In Peter Szendy's discussion on Schoenberg's modern organicism and what he calls "the modernist regime of listening" (ibid.), 'listening' and 'work' collapse into each other, and the problem of the music work can be articulated. The image of the composer and the practice of composition can be understood differently, and by extension, databasing will be reconfigured as well, and the database as that which is a product (or work) of databasing will in turn be seen differently.

**The Work Problem** In what does this articulation of the problem of the music work consist of? First of all, why is it a problem? As Szendy suggests with the metaphor of the the self-amputation of the listener, we as the body of listeners (the listening body) would be severed. Put differently, listening itself would be delineated from outside itself. That is to say, with the presence of an object (music work) which, in its interest of perfecting, polishing, and thus giving itself a 'finish,' would shape and reshape listening until an ideal listening was achieved. The work would work out listening as work. This is the point: the moment the work of music begins to act as 'work' itself, its listening is worked out as well, not by the physicality of the waves in media, nor by the virtuality of perception, but by the concept of 'work' itself. The "modern regime of listening" played this through and through, shaping its listeners into an idealized listener. In this sense, this

listener displays a measured listening, tailored, developed into different degrees of listening like a *gradus ad auscultare*, one existing beyond any psychoacoustic measuring.

**Working Rules** From the shaping of listening by the work, and from the working activity that is performed by the work itself, the presence of the work as an object can be thus traced. In other words, the work of the music work can be considered as the work of an agent in the composition network: the music-work-as-object and the listening-as-object become nodes. In the modernist regime, the hierarchy priorizes the ruler (work-node), and all that can be identified with listening-node is arrived at by subordinating the relations to the work-node, and by restricting the directionality of this relation to being **work** $\rightarrow$ **listening**. The only exception is, of course, the extremely cultivated case of composers, which revert the arrow. This exception, however, is not so much an exception, as it is the prescription of the rule and of ruling itself, since it is this reversal what enables the structure in the first place: the ruling of the exception. This is what occurs when the work begins to listen to itself. Which is radically different from the case of a *listening* that listens to itself. Jean-Luc Nancy, in the foreword to Szendy's text, considers that "music places us outside of ourselves," because "...what listens to itself is not just what resounds in the self and what rebounds to the self: this same movement, and this very movement, places it outside of self and makes its rebound overflow. (ibid., p. xii)" As I have explained before using Nancy's concept of the resonance of a return (See 2.1.2), listening is an approach to the relationship in self. Implementing this relationship within the dynamics of listening and work, the previous graph can be revised as follows. 'Work' and 'listening' would exist as well in relation, with the difference now that it is a relation that exists in a permanent state of overload, redundancy, or excess: **work** $\leftarrow\rightarrow$ **listening**.

**A Space of Difference** In thinking listening this way, the concept of the work is relieved of its duties, discharged, fired, it becomes unemployed. The regime of listening becomes a listening

space, but a space not of equality: a space of difference. Within this space where difference resonates, the music work no longer 'works', it 'unworks' (See 2.1.4). That is to say, the relations between the different resonant points in the composition network expose themselves in a state of suspension, or interruption, creating space with the space of their own incompleteness by the fractality of their fracture. Thus, inoperativity is creation, it is techne, but it is a creativity that is necessarily indefinite, incomplete: the moment it becomes a thing it begins to work in the realm of the 'archi'; the moment it remains suspended upon its limit, it unworks in negation of the 'archi'. One is tempted to place this inoperativity in utopia, in the very instance of the non-place itself, but then one would forget what is already 'there', the fluid medium, as well as gravity itself, which was until recent studies, thought of as unrelated to sound.[16] One would be tempted, equally, to place this inoperativity outside temporality itself, but then one would forget forgetfulness itself. Inoperativity is within the resonant space of an always.

**A Severed Work** What constitutes, then, that moment when the music work becomes a work? How is it possible for the work to become a thing, for the object to become the ruler, for the regime to be built on the first place, if the resonant space is already an inoperative space, interrupted and suspended? I would like to revert Szendy's metaphor of the amputated limb, and propose that it is the music work itself what is amputated, what falls off, the moment that it becomes a finished thing. Thus, just like the modern inattentive listener falls in the uselessness of its excess, sound 'outside' the work is cut from the work, fading out in the uselessness of its excess. Like the human in Kittler's digitally converged apocalypse, redundancy is out of the question, it is left at the gates of the majestic concert hall, with the rest of the (useless) humans: it is literally and conceptually placed outside architecture itself. The created work, in its essential nature of being a cohesive, coherent whole, separates itself from the world of mechanical waves, and forms the one and only

[16]In a recent study, sound itself proven to make (tiny) gravitational fields: "We show that, in fact, sound waves do carry mass —in particular, gravitational mass. This implies that a sound wave not only is affected by gravity but also generates a tiny gravitational field, an aspect not appreciated thus far" (Esposito, Krichevsky, & Nicolis, 2019, 8).

work: the piece of music. A 'piece' not because it is in itself incomplete, but because it is the piece of the whole of the work of the composer.

**Absorbption** I would like to add to this worldview another concept brought by Szendy, that of 'absorption.' He claims that it is the absorption of the listener in the work what is the ultimate aim of this modern regime of listening. Not surprisingly, 'absorption' is the key concept in Iannis Xenakis' narrative of the fours stages of degradation of Western Music's "outside-time structures," article *Towards a Metamusic* (1967): "we can see a phenomenon of absorption of the ancient enharmonic by the diatonic. This must have taken place during the first centuries of Christianity, as part of the Church fathers' struggle against paganism and certain of its manifestations in the arts…" Later, referring to larger structural groupings: "this phenomenon of absorption is comparable to that of the scales (or modes) of the Renaissance by the major diatonic scale, which perpetuates the ancient syntonon diatonic…" Finally, "one can observe the phenomenon of the absorption of imperfect octaves by the perfect octave by virtue of the basic rules of consonance" (Xenakis, 1992, pp. 189–190). The final stage of this process of absorption and degradation comes with atonalism, which "practically abandoned all outside-time structure" (ibid., p. 193). However, Xenakis' narrative contextualizes his sieve theory, devised as a means to "establish for the first time an axiomatic system, and to bring forth a formalization which will unify the ancient past, the present, and the future" (ibid., p. 182). Further, Xenakis formulated this theory with computers in mind, that is, with its concrete application in computer programs, under the subtitle "suprastructures" (ibid., p. 200). Therefore, considering the organicity of the work in Xenakis's overly modern gesture towards unity, metastructure, and mechanization, which was built in reaction to the "poison that is discharged into our ears" as he witnessed the "industrialization of music [that] already floods our ears in many public places, shops, radio, TV, and airlines, the world over" (ibid., p. 200), how can these notions of inoperativity be found together within architecture? How would this archi-techne be designed? Would it still be the product of the 'archi'? Where is the poison

that Xenakis the architect and composer, was identifying with 'industrialized' music? Is it not a product of modernity itself, as the working that listened to itself to the point of working out a Xenakis-listener-node to the extreme?

### 2.4.3 The Composer As Navigator

> I am motivated to present this architecture, which is linked to antiquity and doubtless to other cultures, because it is an elegant and lively witness to what I have tried to define as an outside-time category, *algebra*, or structure of music, as opposed to its other two categories, in-time and temporal. [emphasis added] (ibid., p. 192)

> With [the relational] model any formatted data base is viewed as a collection of time-varying relations of assorted degrees... this collection is called a *relational algebra*... a query language could be directly based on it... The primary purpose of [relational] algebra is to provide a collection of operations on relations of all degrees... suitable for selecting data from a relational data base. [emphasis added] (Codd, 1972, pp. 1–5)

**Querying the Sieves** If we consider pitches as an outside-time (relational) database, one way of understanding Xenakis' sieve theory is as a query method, for which E.F. Codd's Relational model would fit perfectly. The nature of this consideration stems from the application of algebra as a programmable selection mechanism or simply, filters. Both concepts (sieves and relational algebra) have a common link, which is, not surprisingly, the computer itself, and not just any computer, the IBM-7090.[17] While Xenakis' experiments were carried out on the IBM-7090 mainframe computer located at IBM-France in Paris, Codd himself worked at the IBM Research Laboratory in San Jose, California. Furthermore, this same computer was used by Hiller and Baker in their realization of MUsic SImulator-Interpreter for COMpositional Procedures (MUSICOMP), a pioneering language for algorithmic composition (Ariza, 2005a, p. 44). Most important, the IBM-7090 used

[17] Among other things, the IBM-7090 computer was used in the computation of the first 100,000 digits of $\pi$ (Shanks & W.jun. Wrench, 1962), Roger Shepard's computation of the homonymous 'shepard' tone (N. Shepard, 1964), Alexander Hurwitz's computation of the 19th and 20th mersenne prime numbers,[18] and Peter Sellers' plot-twisting moment in Stanley Kubrick's "Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb:" https://en.wikipedia.org/wiki/Dr._Strangelove

the programming language FORTRAN IV, as can be seen by the printed FORTRAN routines for Xenakis' 1962 work *Atrées (ST/10-3 060962)* (Xenakis, 1992, p. 145).[19] Xenakis's work on sieves came a few years after his experiments on the IBM-7090, and his sieves program was written in Basic and then in C. However, the experience with FORTRAN IV at the IBM-7090 serves nonetheless as a common ancestor to both Xenakis and Codd. For example, Xenakis' transcriptions in early CAC systems were performed with tables of outputted computer data. Further, Ariza (ibid.) writes how "the early systems of Hiller, Xenakis, and Koenig all required manual transcription of computer output into Western notation. The computer output of these early systems was in the form of *alpha-numeric data tables*: each row represents an event, each column represents an event parameter value" [emphasis added] (ibid., p. 94). In this sense, performance in CAC meant interpreting results out of a database.[20]

**Sound Synthesis Parenthesis** (Before continuing, a sound synthesis parenthesis must be opened. While Xenakis praised the speed at which the IBM-7090 could perform computations, Max Mathews (Mathews, 1963), then director of the Behavioral Research Laboratory at Bell Telephone Laboratories, wrote:

> A high-speed machine such as the IBM-7090, using the programs described later in this article, can compute *only about 5000 numbers per second* when generating a reasonably complex sound. However, the numbers can be temporarily stored on one of the computer's digital magnetic tapes, and this tape can subsequently be replayed at rates up to 30,000 numbers per second (each number being a 12-bit binary number). [emphasis added] (ibid., p. 553)

Mathews' concern for speed was grounded on the need to achieve sound synthesis, which meant fast computations of the sample theorem. Initially, the first synthesized sound was obtained in 1957, with the (assembly-code written) MUSIC 1 program with the IBM-704 (a predecessor of the IBM-7090). Later, when Bell Labs obtained the IBM-7094 —which "was a very, very effective

[19] Interestingly, given that Ariza finds Xenakis' sieves code unusable (ibid., p. 1), chances are that the printed code for the ST/10-3 composition is likewise useless.

[20] For further reference on the early uses of computers in CAC, I refer the reader to Ariza's PhD thesis (ibid.).

machine" (C. Roads & Mathews, 1980, p. 16)—, and in combination with the (then) widely available FORTRAN compiler, Mathews could develop the MUSIC I program, into MUSIC V, which became the first portable computer music language designed for computer music synthesis.[21] I will close this parenthesis, not without returning to this discussion in the following section (See 2.4.4))

**Algebraic Abstractions for Freedom** Xenakis' and Codd's papers came out around the same time: Xenakis' english publication of *Towards a Metamusic* was in 1970, Codd's papers were published in 1970 and 1972. While sieve theory was aimed at providing a plethora of computable sets (or relations) of pitches, according to different temperings of the smallest displacement unit and the selected value for the modulo operator, Codd's relational algebra was meant the internal structure of a query language for selecting elements based on their relations. Both of these can be considered algebraic abstractions of a selection process. In the case of Xenakis, the abstraction was one held outside-time. This meant that the composer could make a snapshot, or a tomography of the pitch space in order to analyze it, extrapolating structural relations. In Codd's case, the abstraction was spatial: the query language would be separated from the database itself, allowing a distance between a 'backend' and a 'frontend,' allowing databasers to perform queries without worrying about internal data structures, memory allocation, since these operations would occur in the background. Both methods came as an extension of freedom on the human operator: by black-boxing hardware-specific programming, the human operator could devise any kind of algebraic queries, thus operating at a higher level of abstraction, enabling a less problematic kind of envisioning. Conversely, Xenakis writes: "freed from tedious calculations, the composer is able to devote himself to the general problems that the new musical form poses, and to explore the nooks and crannies of this form while *modifying the values of the input data…*" [emphasis added] (Xenakis, 1992, p. 144).

[21] As an example, I would refer the reader to James Tenney's work from 1962 "Five Stochastic Studies," which can be found on a YouTube account on his name: https://www.youtube.com/channel/UCEzSaoPnxCJVzXxA9obuRWg/videos. Roads, while interviewing Matthews recalls this piece to be named "Noise Studies" (C. Roads & Mathews, 1980, p. 18), which fades out the reference to Xenakis' music.

**A Cosmic Vessel and an Armchair** Therefore, the composer delegates to the computer the minutiae of arduous iterative computations: precisely what the computer is better at than the human. As a result, in Xenakis' view, and in resonance with programmer Charles Bachman's claim for the *The Programmer as Navigator* (Bachman, 1973), the composer became a pilot:

> With the aid of electronic computers the composer becomes a sort of pilot: he presses the buttons, introduces coordinates, and supervises the controls of *a cosmic vessel sailing in the space of sound*, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream. *Now he can explore them at his ease, seated in an armchair* [emphasis added] (Xenakis, 1992, p. 144)

Codd's and Xenakis' propositions were abstractions deeply rooted in and contextualized against a backdrop of their own fields. Xenakis wrote against the current state of Western Music with its "degradation of outside-time structures", the "followers of information theory" and the "intuitionists." Codd wrote against the previously developed hierarchical and network database models. Most important, these tools and their development had the human operator's considerations in mind. The composer, like the databaser, would engage in a rudimentary and limited, but still present, feedback process at the *input* level. That is to say, unless rewriting the code, which consisted in a very long and economically expensive process combining punch cards and magnetic tapes, the composer and the databaser could change the input several times, achieving different outputs in a matter of hours.[22] For example, queries made on the relational model would appear on screen at a very fast rate, thus enabling better tuning of the input in relation to a wanted output. Likewise, the composer could modify the input values to highly complex calculations that would otherwise take a long time, or be error prone. The limitation, of course, is the level of intervention with the code itself, which the overall circuitry would itself complicate; criticism on account of this shortcoming of the circuit would thus be rendered anachronic, but recalling these limitations places composition and databasing in perspective.

[22] As a reference, the computation of the first 100,000 values of $\pi$ took about eight and a half hours (Shanks & W.jun. Wrench, 1962).

### 2.4.4 The Database As Performer

I would like to take an improvisational turn that would make Xenakis fall off his armchair. Xenakis' fall would be contemplated against the spirit of the later discussions on interaction that came with George Lewis and *Voyager* (G. Lewis, 2000; G. E. Lewis, 1999; Rowe et al., 1993). Lewis called his approach "a improvisational, nonhierarchical, subject-subject model of discourse, rather than a stimulus/response setup" (G. E. Lewis, 1999, p. 104). Thus, the activity of the composer was reconfigured in networked relation *with* the computer. That is to say, Xenakis' metaphor of the computer as pilot, would be turned upside down, altogether reconfiguring the navigational metaphor: the ship begins to navigate itself.

**The Computer as a Musical Instrument** It is now pertintent to bring back Max Mathews "computer as musical instrument" (Mathews, 1963). The architecture of MUSIC-V is buit on concept of the computer as an instrument that the composer performs by providing it a score. The three stages of data flow (reading, sorting, and executing) are modeled from three music concepts: score, metronome, and instrument. It is significant that two (human) elements of (european) music tradition (composer and performer) are missing from this triad, as well as the programmer. (A fourth missing element, the improviser, will appear further down this text.) On the one hand, the hybrid musical instrument that the computer represented already collapsed three concepts into one, resulting in a hybrid score/metronome/instrument. On the other hand, it can be argued that by this ellision, the three missing human terms have collapsed into one another, forming a new hybrid definition of composer, performer, and programmer. In any case, this hybridity became evident in the music work itself, as can be read from a rethorics of control in favor of the composer:

> So far I have described use of the computer solely as a musical instrument. The composer writes one line of parameters for each note he wishes played and hence has complete control of the note. He is omnipotent, except for lack of control over the noise produced by the random-number unit generators. *Here a minor liberty is allowed the computer.* [emphasis added] (ibid., p. 557)

**A Minor Liberty** As can be read at the end of the introduction to MUSIC-V (ibid.), the extent of this "minor liberty" was measured against Hiller and Isaacson's previous work (Hiller & Isaacson, 1959), which Mathews describes as an extreme case of the computer as composer: "the computer can be given a set of rules, plus a random-number generator, and can simply be *turned on* to generator any amount of music" (Mathews, 1963, p. 557). On the one hand, Mathews' argument is based on the "omnipotence" of the composer in front of the computer. Control of the music work is not something that can be delegated to the computer, unless it comprises lengthy calculations of pseudorandomness. On the other hand, as Ariza has shown, the computer output of early CAC has been often misconceived in the literature as directly musical output, disregarding the extensive transcription work on the part of composers (Ariza, 2005a). Nonetheless, Mathews' "minor liberty" can be considered as a reassurment for the reader that computers would not take control over music, let alone over the world. As I see it, arguing for control while granting some liberty relates to a negotiation between composer's work and computer time. Because of the correlation between sonic complexity and parameter input, "the composer must make his own compromise between interest, cost, and work" (Mathews, 1963, p. 555). Pseudorandom generators introduced complexity in an efficient way (Cámara Halac, 2018a). Therefore, in an economical choice, arguing for omnipotence allowed for some aesthetics agency to come from computers.

**The Computer as a Player** Rowe (ibid.) identified two paradimgs within interactive systems: *instrument* and *player*. The instrument paradigm comprises systems in which performance gestures are sensed (collecting gestural data), processed (reading and interpreting data), and a response (sonic output) is elaborated. The player paradigm comprises the creation of "an artificial player, a musical presence with a personality and behavior of its own…" (ibid., Chapter 1). Therefore, the instrument itself contains embedded processes that grant some level of independence. This means that the composer intentionally relinquishes control of the artwork's structure to the system itself. Like Vaggione's concept of the computer as a complex system in which the composer "is

imbedded in a network within which he or she can act, design, and experience concrete tools and (meaningful) musical situations" (Vaggione, 2001), the human node breaks the traditionally hierarchical structure of composer-work. What this amount to is a distributed authority of the work among the elements of the system. For example, in *Voyager*, "the computer system is not an instrument, and therefore cannot be controlled by a performer. Rather, the system is a multi-instrumental player with its own instrument" (G. E. Lewis, 1999, p. 103). The computer becomes an improvisation partner. While the limitations of computer capabilities precluded more complex conceptualizations of the type of interactivity between computer and composer in MUSIC-V, as personal computers became affordable the type of negotiations no longer depeded on economic decisions. For Lewis, this negotiation existed sonically between computer and improviser:

> There is no built-in hierarchy of human leader/computer follower, no 'veto' buttons, pedals, or cues. All communication between the system and the improviser takes place sonically. A performance of Voyager is in a very real sense the result of a process of negotiation between the computer and the improviser. (ibid., p. 104)

**Programming Decisions** However, in order to implement concepts coming from artificial intelligence such as machine listening and learning, the complexity of the program itself increases exponentially. In light of these difficulties arising from programming, in response to Lewis' criticism of the Max patching paradigm rooted on trigger-based interactivity, Miller Puckette responds: "If you wish your computer to be more than just a musical instrument —if you want it to be an improvisation partner, for instance— you need a programming language. One thing people in this situation might want to do is write Max external C procedure" (Rowe et al., 1993, p. 8). As Rowe writes:

> To arrive at a more sophisticated interaction, or *cooperation*, the system must be able to understand the directions and goals of a human counterpart sufficiently to predict where those directions will lead and must know enough about composition to be able to reinforce the goals at the same moment as they are achieved in the human performance. (Rowe, 1992, Chapter 8)

Furthermore, the player paradigm and its subsequence reconfiguration of authority is possible by means of an implemented database: the computer stores features during the course of the performance, which are then averaged over time, and which serve as guides for the sonic outcome on the part of the computer. As I mentioned earlier, while the guidance of the database provides paths through uncharted territories, it also hides other paths (See 1.3.3). *Voyager* indeed brings interactivity between human and nonhumans to another stage, and because of it, music composition can be seen differently. However, the intricacies of the programming decisions are still in play, specifically in the modelling of musical concepts within data structures.

**Anachronic Composers** This notion of interactivity differs greatly from Xenakis' (modern) composer. He is sitting quietly in his armchair pressing buttons in 1962. By pressing them and inputting certain values, he controls the output, since he knows beforehand the internal mechanisms that are embedded in the software. This image of the modern composer in front of computer technology can also be found in, for example, Edgar Varèse: "The computing machine is a marvelous invention and seems almost superhuman. But, in reality, it is as limited as the mind of the individual who feeds it material" (Varese, 2004, p. 20). Varèse's words, however, refer to the creative limit that a computer might have, which is always a function of the input and, by extension, of material itself. Furthermore, in relation to electronic technology, he writes: "like the computer, the machines we use for making music can only give back what we put into them" (ibid., p. 20). Therefore, from these images of Varese-composer and Xenakis-composer, two axioms can be extrapolated: first, that composers do not lose control of the output; second, that the way to interact with computers is precisely telling them what and when to do it, so that the user is in total operative control. It is against these two axioms of computers and composition that Lewis' work in the late 1980s and 1990s can be contextualized. More precisely, it is because of the anachronic presence of the modern 'eurocentric' composer, and of its popularity among computer music history, that Lewis brings into surface the question of interactivity.

**bang** Placing Max into perspective by commenting on the social and cultural environment of computer music of the late 1980s, Lewis writes:

> 'interaction' in computer music has moved from being considered the province of kooks and charlatans (I'm proud to have been one of those), to a position where composers now feel obliged to 'go interactive' in order to stay abreast of newer developments in the field (Rowe et al., 1993, p. 11).

The way in which interactivity was considered in the 'interactive' music made with Max was, for Lewis, determined by a fundamental feature of program —the 'trigger'—, which, in turn, was grounded on a more general programming concept: the conception of the patching window as a digital equivalent to the analog synthesizer's patching mechanism, where graphic cords are equivalent to cables, equating data flow with voltage flow. Nonetheless, the trigger ('bang') is a feature, not a bug, unless it is used as an extension of the stimulus/response paradigm of interactivity. In other words, in resonance with Vaggione (See 2.3.3), subordinating music events to triggers by a human operator brings out a certain military metaphor which Lewis calls "hear and obey" (ibid., p. 11). This metaphor can easily be extended to that of weaponry itself, and to the unfortunate naming of 'bang' method of objects, a method which (generally) triggers the object's core routine.[23] In order to address this shortcoming of interactivity, Lewis relates it to rudimentary mental processes, or as he puts it, to "amoeba- or roach-like automata" (ibid., p. 11). In this sense, not only interactivity itself is at stake by the presence of a simple model of interaction. For Lewis, the crucial aspect of this model is the empowering of the image of the composer. This intentionally (very) simple automaton promotes two fundamentally hierarchical notions that Lewis attempts to deconstruct. On the one hand, the composer as controller who would never relinquish control of the music work, that is, the modern (eurological) image of the composer, and the old ghost train that comes with it: "The social, cultural, and gender isolation of the computer music fraternity (for

[23] However unfortunate this 'bang' name is, it makes one think back to the 1946 setting of the UNIVAC computer, in the military context of the Manhattan Project, for which the computer was used to get closer to the 'H' bomb. That is to say, even if 'bang' was named differently, the computer itself would be inevitably linked to this particularly *big* bang.

that is what it is)" (ibid., p. 11). This image leaves improvisation, together with non-eurological thinking out of the scope of contemporary music research. On the other hand, the human operator, as the higher (architectural) mind that would not allow for the nonhuman to become an operational agent beyond the instructions for which it was designed. In this sense, the simple-level automaton is a symbolic restrain representing the classical concept of the human itself, which allows a non-threatening relation between man and machine that can be considered functional, productive, and operative.

**Nonhuman composers** One is tempted to claim that the first of these images —the reified composer— is determined by the second —the reified human—, and that their relation is a matter of depth or inheritance. Thus, in order to redefine the composer one would have to redefine the human. In turn, this depth would be measured against that which is nonhuman, and by extension, that which is non-composer. In Lewis' narrative, this entails the redefinition of composition itself by making the non-composer (e.g., what was eurologically considered the 'improviser' or the 'performer') resound back into composition, regrouping the concept 'composer' itself, but not as a whole, since now the extent of its terms have found places within a networked system. This is precisely what he does in *Voyager*. The composer, like the human, became regrouped in hybridity. A hybridity that cannot be considered 'on its own', since it escapes any idea of ownness (or oneness). Therefore, a hybridity that is expanded in networked resonance. It is in this sense that Lewis' proposal is geared towards an interactive (computer) music *not entirely* driven by input.

## Fractured Works

> The composer therewith relinquishes some degree of low-level control over every single bloop and bleep in order to obtain more complex macrostructural behavior from the total musical system. The output of such entities might be influenced by input, but *not entirely* driven by it. [emphasis added] (ibid., p. 11)

It is precisely this 'not entirely,' as a negation of wholeness, what begins to question

the basis upon which our general concept of the human is built, and by extension, the agency of everything that falls outside of its definition. It is the beginning of a breakage, a crack on the foundation of Xenakis' (old) armchair, from which the state of suspension of the concept of the music work can be understood:

> With this in mind, it becomes easier to see that Voyager is *not really a 'work'* in the modernist sense —heroic, visionary, unique (Foster 1983). Rather, I choose to explore allegory and metatextuality, the programmatic, the depictive— and through embedded indeterminacy [pseudorandom generators], the contingent. Ultimately, the subject of Voyager is not technology or computers at all, but musicality itself. [emphasis added] (G. E. Lewis, 1999, p. 110)

Furthermore, what this fracture reveals is hybrid nature of the notion of what is real and what is virtual. Understood traditionally, or better, understood under the stipulations of the first wave cyberneticians, the composer, being the real factor in the constitution of the (modern) image of the composer, is faced with the virtuality of the computer. Upon this encounter, the virtual comes as a form of threat to replace that which is real. In this sense, this is how I would like to approach Lewis' consideration of *Voyager* as "not really a work." I hope the reader would forgive me for having borrowed these adjectives out of context —'entirely' and 'really'— so as to allow my argument to echo with Lewis' for a while. On the one hand, as Lewis claims, the goal of the interactivity between the composer and the computer is to allow the real and the virtual, "virtuality and physicality," to engage in the production of a hybrid that "strengthens on a human scale. Seen in this light, virtuality should enhance, not interfere, with communication between us" (ibid., p. 110). Therefore, considering the role of virtuality after new media integrated theories of embodiment, the computer reveals to the human —composer, improviser, performer— the very condition of its own virtuality, that is, virtuality itself within the human. In the case of *Voyager*, this virtuality is sonic, it comes as the "emotional transduction" that Lewis aims for with this computer system. Therefore, it can't be 'really' a work, because it is virtuality itself resounding back. Another way to approach this is the fact that the computer can be said to be 'listening' to

the performer, given that its real-time analysis is content-based, using techniques that have been applied to music information retrieval over the years (See 1.3.1).

**Databasing Vessel** Understood as a listener, *Voyager* engages not only with signal processing at the lower level, it engages with the resonant process of the relation to self. Furthermore, the computer is not only listening, it is *databasing*, because it is keeping record of the listened features, and in so doing, it becomes empowered with the database itself. This database of actions, however, is the sonic trace of the performance itself, which is what is most surprising of its agency, and what resounds most in time. Therefore, far from being 'really a work', but also far from Lewis' notions of narrative in the sense of "allegory and metatextuality, the programmatic, the depictive" (ibid., p. 110), I consider *Voyager* an unwork of music, one that puts into question —though, to a certain extent— the operativity of the music work itself. To a certain extent, because the notion of productivity and cohesion are still present within Lewis' music and texts, and also, to the (paradoxical) extent that it is still a 'work,' a destiny that somehow manages to persist within the practice of composition. Nonetheless, and without a doubt, Lewis' claim for the "non-eurocentric computer music" (ibid., p. 107) can be a starting point to the conceptualization of the unwork.

### 2.4.5 The Severed Object Of Music

> [The] Heideggerian 'work of art' is able to present a unified picture that may be used for political purposes [it] is only what it is in the world that it opens. . . Nancy is seeking a 'workless' or 'unworking' work, *a work that refuses to create itself as a total work*. Hence, Nancy proposes an artwork that would offer itself as a permanently open whole, the concept of art remaining undecided and lacking anything that might unify it. [emphasis added] (Gratton & Morin, 2015)

**An Incomplete Object** I would like to refer once again to Jean-Luc Nancy's concept of inoperativity (See 2.1.4), this time in relation to the music object. I argue that, given that the inoperativity of the listening experience reveals itself as the interaction between resonance —as the *différance*

within sense and sensuality— and the unworking of the network, its resulting object, instead of being a complete whole —a finished, integral 'thing', or even, a 'piece'[24]—, it becomes a severed music object. This object is different from Pierre Schaeffer's music or sound object, which comes to represent material with which to work. Neither it is related to Vaggione's concept of object, which comes from object-oriented programming, meaning every composable primitive, from the micro to the macro. In both of the above, the object is used to provide, though not without their author's intervention, a notion of *coherence* to the work.

**Remains of Listening** The object I am referring to resides in memory, as the remains of the event of an exposure. It is inherently linked to the fractured way in which our own memory works, and it is impossible to define, since it has no beginning and no end. Its dimensionality includes both beginning and ending simultaneously. This object is the spectral evidence of a musical event, or better, of the happening that takes place in listening. In being evidence, it becomes subject of analysis, it is forensic. In being fractured, it is the evidence of a destruction. In being severed, and this is the central aspect that I would like to focus on, risking simultaneously the severing of the object itself, it becomes the evidence of a sacrifice. If it can be said that the music object is a severed object, then the question of its severing necessarily relates to the question of listening. Therefore, by listening —and, by this, I mean entering in resonance with resonance itself, exposing the self to that which returns to itself— I participate in this severing, because in listening I choose what to listen in spite of being already deprived from that choice.

**Sources and Sorcerers** The sounds onstage are always before and after the staging. The severed object of music is what, as listeners, we grab from the stage, what we choose to rip from the sounding waves, and also what we cannot help but feeling so much a part of us before noticing it is happening. Severing is yet another way of thinking the aesthetic experience of listening, but it

[24]Since, the notion of a 'piece' presupposes that of the whole to which it belongs.

is not as passive as it seems. Severing empowers the listener, it is the tool of listening, the reversed stilus, the inverted mouse, the part of the human that necessarily is nonhuman. With it, we can make the world appear, but only as a fraction, because 'it' can never be *completely*. The severed object of music is always severed, but never in the same way, since there are as many severings as there are listeners, and as many listenings as there are moments. In this difference, what is resonant is the object of music, which is never one and the same because it is a singularity resonanting in plurality. Composers have traditionally been considered a 'source' of this object, or better, the one at the door, the key keeper that has access to the door that opens up the flow of inspiration. The composer, but also the programmer with access to the source code, which unless it is opened, is hidden to the rest; and, unless you know the language, it is complete pseudo-linguistic nonsense with weird punctuation marks, sometimes closer to poetry than it is to extreme formalism.

```
#!/bin/bash

# Palabritas que hacen cosas

while true
do
        for ever in rose is a
        do
                say $ever
                sleep $((RANDOM/10000))
        done
done
```

Listing 2.1: Little words that do things

In this access to the source, the programmer and the composer are traditionally kept at a distance, as if their listening were of some other sort, engaging with the very essence of the source, drinking the water from the originary fountain, satisfying an originary thirst. Therefore, if this is the role of the composer and the programer, if this is their relation to the source, then, they are the first to perform the severing. In the hierarchy of the consequent severings, they are at the top.

Further, if they are the first severers, they are the first who perform the first listening. They are the listeners at the top of the mountain, next to the source of all fountains. On the way in and out of the world, the sorcerers of condensation.

> This is why for some years I have experimented with releasing music under other people's names, so as to dilute their style, and under multiple versions of my own name, to case doubt on any claim to the future. (Nilson, 2016)

**Naming** I would like to point out now, that it is not my intention here to sever the head of the sorcerer, because it is an illusion that does not allow me to do so. It is not my illusion, although I have described how I interpret it, and it comes as a product of a reification of the composer, but also of the human itself as the one and only owner of the world —that is, owner of the mountain itself, and of the water, and every particle of the one and only universe. In being in resonance, listeners become the resonant world, that is, the self begins to resonate as space. In this sense, it is the world what is listened to, and it is a world that has no apparent origin. However, the composition —the written score, like the written code— propose their own origin —the composer, the programmer. Thus, they give an origin to the world itself by providing an answer (a name) to the question of creation: Who created this music? *this* composer. The answer, therefore, has a 'this' that comes in the form of the name of the composer. This name becomes attached to the flowing of the source. Therefore, the name of the composer is like a timbre stamp that is applied to the listening experience itself. Further, the severing style itself now can be named. How many differet names or anagrams would it take for Click Nilson's style to dilute or to arrive at the unclaimable work of art? The name of the composer becomes a synecdoche of the source itself, directly naming part of the source. This applies, quite literally in some cases, to the name of the program and the name of the programmer (the 'max' in Max Mathews and the 'smith' in 'msp').[25]

[25] 'Max' is named after the 'father' of computer music Max Mathews, and MAX/MSP contains Miller Smith Puckettes's initials. Friendly gestures, most probably, but also pointers to originary sources, sources of inspiration, historical references that contextualize computer music software within broader social and environmental structures.

**Dynamics** Furthermore, the activity of the sorcerer lends itself to its signature. In other words, the manner in which the composer defines the music, from beginning to end, becomes the shape of the music, understanding 'shape' or 'form' as something that is at once behind and in front of the singularity of the listened music. It is behind, because it is the activity of sound sources —speakers, musical instruments, or simply media in general—, the movement of air pressure. It is in front, because it filters the memory of the activity of sound sources. However, this composed shape and the singularity act together in the moment of listening. The question is, then, regarding the dynamics of this activity. Given that this activity happens during listening, what I addressing now is precisely how the shape of the music interacts with the listening itself. That is to say, the interaction between shape —but also the form, the idea— and the singularity of the listened. Interaction, here, refers to the shared activity that occurs 'inside' listening itself, and it happens 'inside' because of the severing that needed to occur prior —or immediately at— the resonant oscillation of air pressure. This is what I consider the moment of listening that is none other than listening to music. However, once this severing has occurred, and within its momentum, it is the internal dynamics that enter into play, and it is the shape of the music what begins to delineate the shape of the listened.

**Masterwork** Understood in this way, that is, the shape of the music as a force that produces a certain listening experience, therefore, the internal dynamics are already written. The singularity of the listened becomes (almost) one and the same with the shape of the music. 'Almost,' because it is not that the listened brings no resistance to this ideal force. The singularity of the listened is resistance itself, like I have mentioned before in relation to the trace (See 2.2.2). It acts as resistance itself, and its force is not enough to resist the command of the excellent work. This is the very presence of the masterwork, at work, the work of a master that requires the slave —a slave that is not the rest of the works but the outshunned singularities that have been muted by its very own presence. 'Almost,' in the hope that its work can be relativized, disarticulated, disentangled

from the source of sources, brought down the stream to the place where singularities can resonate in endless forms of matter. However, the problem is now of a different sort. Even if resisting forces match those of the masterwork, then, like Derrida's concept of a paralysis of memory, we can encounter a paralysis of listening itself. This paralysis. This might (also) be what Szendy means, as well, by the cutting loose of the inattentive listener in modernity, but in a different way. It is not a paralysis caused by distraction, it is a paralysis caused by the very force that is needed to match the force of the master work. It is a paralysis that is directly called for from outside —from the shape of the music itself—, one which prevents any further listening. This is what is called for by the work of the masterwork: pure —and utterly ideal— silence.

**Architecture of Obedience** Therefore, within these dynamics of work, what results is a function of the predicates, it is the architecture of obedience that is written in the form of a music work, with the one and only aim which is for it to 'work.' Thus, the composer engaging with this dynamics of working out the work, of creating the structures, becomes the architect of the listened, the creator of a listening that of which he himself is the only chief. The sorcerer in charge of quenching a thirst that is only there because it is always already there, beforehand, instantiated with its own creation. The question now is how can this dynamics be approached once that I have recognized that it is there. How can composition continue, a composition that does not participate in this dynamics? A composition that is not a force? A composition that is not 'really' or 'entirely' a composition? A composition that does not impose its shape? A music work that is not a work but that still resonates within listening?

### 2.4.6 Anarchy And The Unwork

What characterizes the aesthetic dimension in the severed music object of the composition that does not impose its own listening is inoperativity. In this sense, the practice of music composition can be understood in terms of Nancy's positive, active force of unworking. The condition of

unworking in relation to works of art is exposed by a certain resistance present in the unwork of art. This resistance is a force of interruption and suspension that prevents the notion of a whole to reach completion. The case is quite different from that of the 'open' work, since the work never reaches completion.

**Place in Common** The unwork radically differs from the notion of an open work as is the case, for example, of Umberto Eco's famous formulation that "the work of art is a complete and closed form in its uniqueness as a balanced organic whole, while at the same time constituting an open product on account of its susceptibility to countless different interpretations…" (Eco, 2004). Instead of openness being located in the interpretation, the opennes in inherent to the hybridity of its construction. The construction, in turn, is a result of the reticulated and fragmented state of exposure between the human and the nonhuman. In this sense, the limit of the unwork is the exposure of exposure itself, that is, an instantiation of the place in common.

**Disintegrated Imperative** I would like to analyze the inoperativity of the music in relation to the dynamics of the shape of the unwork and the singularity of the listened. The former, in being a disintegrated imperative —i.e., without the integrity that is required of the imperative for it to work as command and instruction—, cannot behave as a force in its own right. This is not to mean that it 'fails' as a force, for if this were the case, such failure would be its paradoxical success. At this point it would be useful to revise Kim Cascone's consideration of the aesthetics of failure (Cascone, 2000). In his analysis of the 'post-digital' culture of the late 1990s, Cascone identified electronic music outside academia as one related to the unintended uses of computer music software, also known as glitch art:

> It is from the 'failure' of digital technology that this new work has emerged: glitches, bugs, application errors, system crashes, clipping, aliasing, distortion, quantization noise, and even the noise floor of computer sound cards are the raw materials composers seek to incorporate into their music. (ibid., p. 13)

**Blind Experimentation** Within what he called the "cultural feedback loop in the circuit of the Internet" —where artists engage with download and upload of software tools and artworks— Cascone describes a 'modular' approach regarding music creation as being grounded in the use of (recorded) samples and later mixing (ibid., p. 17). His argument is that "electronica DJs typically view individual tracks as *pieces* that can be layered and mixed freely" [emphasis added] (ibid., p. 17). In atomizing this use of samples, glitch art descended to the micro-level, but precisely by this descent, it sacrificed the whole for the parts, that is, it became a case of extreme modularity that "affected the listening habits of electronica aficionados" (ibid., p. 17). Therefore, Cascone's conclusion is to call for new tools "built with an educational bent in mind" (ibid., p. 17), bridging the gap between academic and non-academic electronic music, and therefore illuminating glitch music "past its initial stage of blind experimentation" (ibid., p. 17).

**Doctoring the Glitch** It must be noted that Cascone's inclination towards bringing academic knowledge to the academy of the Internet refers not only to computer music software. Professors, generally of computer music, in several universities across the USA have been openly uploading class materials, patches, softwares, and many other highly useful technical information; not to mention the free online publishing of conference proceedings that have spawned in the last 20 years. Cascone's rendering of this educational turn can be understood with an authoritative and dated tilt on his end. Particularly, consider what he writes in relation to the form of glitch music, which is the last arguing moment before his claim for education:

> But it seems this approach affects the listening habits of electronica aficionados... the 'atomic' parts, or samples, used in composing electronica from small modular pieces had become the whole. This is a clear indication that contemporary computer music has become fragmented, it is composed of stratified layers that intermingle and defer meaning until the listener takes an active role in the production of meaning. (ibid., p. 17)

**Unnecessary Blindfolds** How are we to interpret Cascone's call for education? What is the center of this education: music technology, composition, or listening? If fragmentation, modularity, stratification, and deferred meaning are 'affecting' listening habits, are these 'habits' themselves that need to be taught? Or is the structure of the music in desperate need of medical attention? These ambiguities in his argument, however, I chose to understand as coming out of the main premise of the text, that of extending the concept of failure from the technology itself to the analysis of the work. Thus, in Cascone's view, the aesthetics of failure of the late 1990s is still 'failing' to enter academia because it is itself 'failing' to achieve the same standards of formal cohesion that are required by the modern conception of the music 'work'. Therefore, instead of finding an academic cure for blind experimentalism, I would claim to understand failure itself as an unnecessary blindfold since, at least in my consideration of the unwork, if there is no notion of success in the technology involved, there need not be any in the work itself. The success, if any, exists within the composer, and as such, it does so in relation to the very goal of disintegrating the imperative. This success is unrelated to popularity, for example, as is the case with software production, in which more users mean generally more chances of survival. This success is unrelated to value, since there is no measuring system that can determine how much of the imperative was disintegrated. In being for the composer, success is, like listening, inevitably private, a personal construction, like any other personal growth, or the overcoming of fears.

**Spectral Remains** The unwork cannot behave like a force, but it can be considered the spectral remains of a force. In this sense, if there is an illusion of a force, it must appear as wreckage, an after dream, a mirror that shows us our skin of the past, the ruins of an empire, or the humidity creeping through the cracks of an old house. However, and this is a big however, these allusions to vessels, to the psyche, to architecture, and to the presence of the past altogether, must be addressed with the same strength as one would address a phantom. The unwork makes us feel the uncanny presence of the past in the now, of the overpowering ghost that brings with it the archontic, in the

shape of our own selves that has been revealed to us as not us, but as yet again us. This is the moment that the unwork carries with it the most crucial aspect of all: it has nothing to give. It gives nothing. And this is when listening finds us without anything to hold on to but our very own resonance. Our very own listening to ourselves listening. The moment where we realize it is our own self that is returning to us. This is our resistance.

**Macroforma** The resonance of a return. This is why the unwork depends so extremely on its very state of fragility: it touches the self from itself, it engages the self with its own touch, with its own skin, with the resonance of itself. The moment this fragility is forgotten is when composers, performers, improvisors, programmers —humans and nonhuman listeners, in the most broadest sense possible— enable an operative `macro` that has a political agency in the shaping of singularities. In order to to provide some insight into the difficulties that arise from this conceptualization of the unwork, I would like to brin again Vaggione. When he writes of the shaping of singularities, he refers to the arbitrariness of the composer. However, he intentionally maintains formal coherence by extending the singularity of a grain (conceptually) to the singularity of a work. Therefore, in expanding this singularity he is ultimately arriving at a very unique and delimited shape that is the work. The contradiction I see here is that, in an attempt to propose a bottom-up approach in which, like Lewis' work, local actions percolate up to global behavior, Vaggione grants his work with an inevitable global behavior that is extremely operative: Vaggione himself. Without a doubt Vaggione (self) *is* singularly, and the value of his music is not put into question. I bring this as an example, as I have mentioned before, of the name of the composer and its impression on the music. In this case, the singularity that is the composer impresses its own singular shape, its own style, its own trace, on the music, and makes it a work. The problem is that the work now engages with its own operativity, with its integrity, and begins to dictate the shape of its own listening.

**Overfitting** Defining anarchy as a paradoxically productive force —a form of destruction which "produces the very thing it reduces" (Derrida & Prenowitz, 1995)—, Derrida locates it at the core of the concept of the archive (See 2.2.3). As I have outlined before, databasing and composition bring forth their relation to the archive, and by doing so, they reveal themselves as repositories for the the archontic principle: bound to the origin and the rule. Like the name of the composer which is written in the shape of the music, the database has too the potential of becoming a source. Databasing becomes an activity of this source, and thus embeds the databaser with a specter of authority (See 2.3.4). Claiming, therefore, that composition can be identified with databasing means translating the 'archic' not only to the performativity of composition, also to the product of composing, to the composer and the composed, to the shape of the music, and to the singularity of the listened. An unwork, therefore, would be a necessarily an-archic work. It is still a work, however, in the sense that it demands from the composer, from the databaser, and from every node in the scope of its network, an incessant operativity. That is to say, the 'un' in unwork does not come from inactivity, from passivity, from an escape of any form of action. Quite the contrary, it is a result of the constant impression of the work, the accumulated efforts towards the 'un' of the thing. An extreme operativity that goes beyond the threshold of its own making so that it reaches a point of inflexion, a bent, an overflow. There is a point in statistics where learning algorithms, given a data set, tend to adapt themselves too closely to the data set, thus failing to render future predictions reliably. This is known as overfitting. Despite its uselessness (or better, because of it) I believe this to be a suitable metaphor for the pursuit of the unwork: precisely by overworking the work, one can find some insight into the 'un,' and thus, one can begin to approach the anarchic in music composition. However, this approach comes not without its warnings, since it means at once, to eradicate the archic with the 'an', which means to introduce a bug in the oedipal loop that could result in unheard-of musical behaviors.

## 2.4.7 [Wip] Work In Progress

```
// code for the "working" pd class.
// it does nothing.

#include <stdio.h>
#include "m_pd.h"

t_class *working_class;

typedef struct working {
        t_object  *x_obj;
        t_symbol *work
        union {
                t_symbol *product;
                t_symbol  *music_piece;
                t_symbol  *music_work;
                t_symbol  *opera;
        } music_work;
        t_symbol  *something_done;
        t_float  *physical_labor, *skill;
        t_atom  *the_work_of_an_author, *oeuvre;
        t_symbol *the_operativity_of_the_composer;
        t_atom  *matrix_operations;
        t_symbol *operetta, *opera_prima, *obra, *open_work;
        t_symbol *a_work_of_art;
        t_symbol *artistic_creation, *techne;
        t_float  *fullTime, *partTime;
        t_symbol *clockwork, *officiate, *office, *act;
        t_symbol *produce, *make_it_work;
        t_float  *magic_work, *work_of_angels;
        t_symbol *blueCollar, *whiteCollar, *slavework, *masterwork;
        t_symbol *Work_as_in_the_application_of_forces;
        //V:"But applied to whom?"
        t_symbol *working_a_field;
        t_symbol *the_internal_workings_of_structures;
        t_symbol *work_in_an_app, *worked_out;
        t_symbol *work_your_hat_off, *workflow, *workspace;
        t_symbol *working_for_food, *hardworking, *labour, *giving_birth;
        t_symbol *all_that_is_remunerated_after_efforts_have_been_given;
        t_symbol *achieve_a_goal, *your_task, *to_work_to_live;
        t_symbol *to_have_a_working_body, *functioning;
        t_symbol *operative, *working_like_a_bee;
        union {
                t_symbol *like_a_bee;
                t_symbol *like_a_member_of_the_hive;
                t_symbol *like_an_ant;
                t_symbol *like_a_worker;
                t_symbol *like_a_coworker;
                t_atom *organized_labour;
```

```
        } workers_union;
        char   work["for","to","after","by"];
        unsigned char *hours;
        t_symbol *working_as_an_extension_of_truth_as_well_as_lies;
        t_symbol *out_of_work, *at_work, *work_in_progress;
        t_symbol *working_for_the_man, *freelancing, *working_under_the_table;
        t_symbol *working_past_a_deadline, *working_in_pairs;
        t_symbol *teamwork, *collaborate, *co-operate;
        t_symbol *paperwork, *networking, *prototyping, *worked-up;
        char  *work_the_crowd, *work_the_system;
        t_symbol *work_a_miracle, *work_your_workers;
        t_symbol *social_worker;
        t_float  *a_ship_works_in_a_heavy_sea, *work_the_levers;
        t_float  *work_for_Facebook, *future_work, *framework;
} t_working;
```

Listing 2.2: Pure Data working class

# Conclusion

. . . placeholder for conclusion abstract . . .

# Appendices

abstract of appendices

## DIANA: Database for Image and Audio Navigation

I use William Brent's `timbreID` —timbre description algorithms— and Antoine Villeret's `pix_opencv` —image descriptors using Computer Vision algorithms—, to develop a new software library for Pure Data. My model consists of a joint Database structure for Image and Audio descriptors suitable for realtime navigation. At its core, the Database is generated by calculating derivatives between both data sets, and it is performed by applying random probabilities, markov chains, or chaotic generators to this navigation. This allows for multiple paths to be traced on each navigation.

### A Database Model

A detailed description of the image and audio navigation system...

> Just as a fractal has the same structure on different scales, a new media object has the same modular structure throughout. Media elements... are represented as collections of discrete samples (Manovich, 2001, p. 30).

> First, data is sampled, most often at regular intervals, such as the grid of pixels used to represent a digital image. The frequency of sampling is referred to as resolution.

> Sampling turns continuous data into discrete data... Second, each sample is quantified, that is, it is assigned a numerical value drawn from a defined range (such as 0-255 in the case of an 8-bit greyscale image) (Manovich, 2002, p. 28)

I define the points in common between Database Practice and Music Composition. I describe the main technical concepts behind Database Navigation and provide use cases from both appendices A and B, the former relating to joint image and audio databases, and the latter to text databases. I then reflect on the quality of this navigation in relation to the type of navigation and results that they obtain.

I use computer vision literature to briefly introduce and describe the most common visual descriptors. I focus on certain descriptors (TBD) which are suitable for live multimedia use, and which I will implement in Appendix A.

I use Timbre Analysis literature to briefly introduce and describe the most useful audio descriptors. I take William Brent's TimbreID library, complementing it with Tae Hong Park's dissertation on timbre recognition, and I focus on the most useful descriptors for live multimedia use (TBD), which I will implement in Appendix A in relation to the image descriptors introduced above.

# ABBY: An Online Environment for Annotated Bibliographies

In order to write this dissertation, I have developed "Abby" an online Text Database tool namely to build an annotated bibliography. The program is mostly written in Javascript, with the data navigation and programming hosted in Github, and the datasets stored in the Google account that New York University has provided me. The annotated bibliography is available at https://fdch.github.io/abby, and the code can be accessed or cloned from https://github.com/fdch/litrev.

## A Text Database

A detailed description of the text database model. . .

# Glossary

**Adaptive Server Enterprise** SAP ASE, originally known as Sybase SQL Server, and also commonly known as Sybase DB or Sybase ASE, is a relational model database server developed by Sybase Corporation, which later became part of SAP AG. ASE is predominantly used on the Unix platform, but is also available for Microsoft Windows. See also: https://en.wikipedia.org/wiki/Adaptive_Server_Enterprise. 38, 166

**Apache** Lauded among the most successful influencers in Open Source, The Apache Software Foundation's commitment to collaborative development has long served as a model for producing consistently high quality software that advances the future of open development.. 38, 166

**Apache Software Foundation** an American non-profit corporation to support Apache software projects, including the Apache HTTP Server. The ASF was formed from the Apache Group and incorporated on March 25, 1999. See also: https://en.wikipedia.org/wiki/The_Apache_Software_Foundation. 166

**Appache Couch Database** Apache CouchDB is open-source database software that focuses on ease of use and having a scalable architecture. See also: http://couchdb.apache.org/. 26, 166

**Application Programming Interface** In computer programming, an application programming interface is a set of subroutine definitions, communication protocols, and tools for building software. See also: https://en.wikipedia.org/wiki/Application_programming_interface. 74, 166

**ArangoDB** ArangoDB is a native multi-model database system developed by ArangoDB Inc. The database system supports three data models with one database core and a unified query language AQL. The query language is declarative and allows the combination of different data access patterns in a single query. ArangoDB is a NoSQL database system but AQL is similar in many ways to SQL.. 38, 166

**AudioGuide** AudioGuide is a program for concatenative synthesis that I'm currently developing with Norbert Schnell, Philippe Esling and Diemo Schwarz. Work began in 2010 at IRCAM when I was composer in residence for musical research. Written in python, it analyzes databases of sound segments and arranges them to follow a target sound according to audio

descriptors. The program outputs soundfile event lists that can be either synthesized (in csound or Max MSP/Pure Data) or translated into symbolic musical notation.. 69, 166

**Automatic Computer Assisted Design** AutoCAD is a commercial computer-aided design and drafting software application. Developed and marketed by Autodesk, AutoCAD was first released in December 1982 as a desktop app running on microcomputers with internal graphics controllers. See also: https://en.wikipedia.org/wiki/AutoCAD. 133, 166

**Automatic Flow for Materials Discovery** A globally available database of 2,118,033 material compounds with over 281,698,389 calculated properties See also: http://aflowlib.org/. 49, 166

**Boost Software Library** Boost is a set of libraries for the C++ programming language that provide support for tasks and structures such as linear algebra, pseudorandom number generation, multithreading, image processing, regular expressions, and unit testing. See also: https://www.boost.org. 38, 166

**Cassandra** The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. Linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages.. 38, 166

**Center for Computer Research in Music and Acoustics** a multi-disciplinary facility where composers and researchers work together using computer-based technology both as an artistic medium and as a research tool. See also: https://ccrma.stanford.edu/. 59, 166

**Center for Network-Centric Cognition and Information Fusion** The Center for Network-Centric Cognition and Information Fusion (NC2IF) explores the information chain from energy detection via sensors and human observation to physical modeling, signal and image processing, pattern recognition, knowledge creation, information infrastructure, and human decision-making-all in the context of organizations and the nation. See also: https://ist.psu.edu/research/centers_labs/nc2if. 48, 166

**Center for New Music and Audio Technologies** a multidisciplinary research center within University of California, Berkeley Department of Music. The Center's goal is to provide a common ground where music, cognitive science, computer science, and other disciplines meet to investigate, invent, and implement creative tools for composition, performance, and research. It was founded in the 1980s by composer Richard Felciano. See also: https://en.wikipedia.org/wiki/Center_for_New_Music_and_Audio_Technologies. 75, 166

**central processing unit** the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logic, controlling, and input/output (I/O) operations specified by the instructions. See also: https://en.wikipedia.org/wiki/Central_processing_unit. 48, 166

**Centre des Musiques Arabes et Mediterraneennes** The Centre for Arab and Mediterranean Music — Centre des Musiques Arabes et Mediterraneennes (CMAM) — is an institution operating under the authority of the Ministry of Cultural Affairs. It was established on 20 December 1991and its statutes were enacted in October 1994. See also: http://cmam.tn/. 41, 166

**Cluster Weighted Modeling** an algorithm-based approach to non-linear prediction of outputs (dependent variables) from inputs (independent variables) based on density estimation using a set of models (clusters) that are each notionally appropriate in a sub-region of the input space. See also: https://en.wikipedia.org/wiki/Cluster-weighted_modeling. 166

**Cmix** A computer music software designed and developed by Paul Lansky. Belongs to the Music N family, although it was designed for a more specific context of concrete music.. 66, 166

**Coastal Data Information Program** an extensive network for monitoring waves and beaches along the coastlines of the United States. Since its inception in 1975, the program has produced a vast database of publicly-accessible environmental data for use by coastal engineers and planners, scientists, mariners, and marine enthusiasts. See also: https://cdip.ucsd.edu/. 47, 166

**Comma Separated Values** a delimited text file that uses a comma to separate values. See also: https://en.wikipedia.org/wiki/Comma-separated_values. 50, 166

**Common Business Oriented Language** A compiled English-like computer programming language designed for business use. See also: https://en.wikipedia.org/wiki/COBOL. 31, 166

**Composition Path** A music making method with interactive map interface See also: https://sihwapark.com/COMPath. 166

**Computer Aided Design** the use of computers (or workstations) to aid in the creation, modification, analysis, or optimization of a design. See also: https://en.wikipedia.org/wiki/Computer-aided_design. 166

**Computer Aided Software Engineering** the domain of software tools used to design and implement applications. CASE tools are similar to and were partly inspired by computer-aided design (CAD) tools used for designing hardware products. See also: https://en.wikipedia.org/wiki/Computer-aided_software_engineering. 166

**Computer Assisted Music Project** A general purpose composition and performance software environment originally based upon William Buxton's SSSP. See also: Free and Vytas, 1988. 28, 166

**Computer Based Education Research Laboratory** A research center based in the University of Illinois See also: https://en.wikipedia.org/wiki/PLATO_(computer_system). 62, 166

**Computer-Aided Algorithmic Composition** Unlike software for sequencing, mixing, or notation, these systems are often diverse and innovative, breaking with traditional musical paradigms of meter, part, or score. These systems expand compositional resources, and offer diverse models of compositional design. See also: (Ariza, 2005a, p. 1). 29, 166

**Conference/Committee on Data Systems Languages** a consortium formed in 1959 to guide the development of a standard programming language that could be used on many computers. This effort led to the development of the programming language COBOL and other technical standards. See also: https://en.wikipedia.org/wiki/CODASYL. 31, 166

**Content Based Unified Interfaces and Descriptors for Audio/music Databases available Online** a new chain of applications through the use of audio/music content descriptors, in the spirit of the MPEG-7 standard See also: Vinet et al., 2002a, 2002b. 70, 166

**Creative Commons** an American non-profit organization devoted to expanding the range of creative works available for others to build upon legally and to share. See also: https://en.wikipedia.org/wiki/Creative_Commons. 77, 166

**Data Definition Language** A data definition or data description language (DDL) is a syntax similar to a computer programming language for defining data structures, especially database schemas. See also: https://en.wikipedia.org/wiki/Data_definition_language. 29, 166

**data manipulation language** A data manipulation language is a computer programming language used for adding, deleting, and modifying data in a database. A DML is often a sublanguage of a broader database language such as SQL, with the DML comprising some of the operators in the language See also: https://en.wikipedia.org/wiki/Data_manipulation_language. 29, 166

**Database Managemet System** a computer program (or more typically, a suite of them) designed to manage a database, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMS use include accounting, human resources and customer support systems. See also: https://en.wikipedia.org/wiki/Category:Database_management_systems. 24, 166

**Digital Alternate Representation of Musical Scores** The DARMS project started in 1963 by Stefan Bauer-Mengelberg and it is one of the first programming languages for music engraving See also: Brinkman, 1983; Erickson, 1975. 57, 166

**Disk Jockey** a person who plays existing recorded music for a live audience. Most common types of DJs include radio DJ, club DJ who performs at a nightclub or music festival and turntablist who uses record players, usually turntables, to manipulate sounds on phonograph records. See also: https://en.wikipedia.org/wiki/Disc_jockey. 70, 166

**Document Object Model** a cross-platform and language-independent application programming interface that treats an HTML, XHTML, or XML document as a tree structure wherein each node is an object representing a part of the document See also: https://en.wikipedia.org/wiki/Document_Object_Model. 32, 166

**Domain Specific Language** a computer language specialized to a particular application domain. This is in contrast to a general-purpose language, which is broadly applicable across domains. See also: https://en.wikipedia.org/wiki/Domain-specific_language. 50, 166

**Electronic dance music** a broad range of percussive electronic music genres made largely for nightclubs, raves and festivals See also: https://en.wikipedia.org/wiki/Electronic_dance_music. 77, 166

**Entity Relationship** A database model that describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities. See also: https://en.wikipedia.org/wiki/Entity-relationship_model. 166

**Essen Associative Code** Essen Associative Code (EsAC) was developed for one-part music, especially for European folksong databases. The code itself was inspired by the Chinese notation JIANPU and consists of cyphers (meaning pitches related to the declared tonic of the mode) and underlines and dots (standing for rhythmic durations). In addition to the code, several programs have been written for PC to analyze, listen, transpose and represent melodies. Conversions are possible into TEX, PCX, MIDI and other formats. See also: http://www.esac-data.org/. 41, 166

**Essentia** Open-source library and tools for audio and music analysis, description and synthesis. 41, 166

**Exasol AG** Exasol is an analytic database management software company. Its product is called Exasol, an in-memory, column-oriented, relational database management system.. 166

**Experiments in Music Intelligence** "Experiments in Music Intelligence (1984) was developed in order to create an interactive tool for composing.... By applying an augmented transition network parser and an object oriented approach, intervals, through inheritance and message passing, have both local and global impact (non-linear composition)" See also: http://artsites.ucsc.edu/faculty/Cope/experiments.htm (Cope, 1987b). 128, 166

**Extensible Markup Language** a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications-all of them free open standards-define XML See also: https://en.wikipedia.org/wiki/XML. 33, 166

**Feedback Delay Network** "Structures well suited for artificial reverberation. These structures are characterized by a set of delay lines connected in a feedback loop through a 'feedback matrix"' See also: https://ccrma.stanford.edu/~jos/cfdn/Feedback_Delay_Networks.html (**icmc/bbp2372.1994.099**; **icmc/bbp2372.1994.098**; **icmc/bbp2372.2015.048**; **icmc/bbp23722015.05** 84, 166

**FORMES** an object-oriented programming environment for music composition and synthesis. 166

**Formula Translation** a general-purpose, compiled imperative programming language that is especially suited to numeric computation and scientific computing. See also: https://en.wikipedia.org/wiki/Fortran. 166

**Freesound** Freesound is a collaborative database of Creative Commons Licensed sounds. Browse, download and share sounds.. 41, 70, 77, 166

**Frequency Modulation** In telecommunications and signal processing, frequency modulation is the encoding of information in a carrier wave by varying the instantaneous frequency of the wave. See also: https://en.wikipedia.org/wiki/Frequency_modulation. 48, 166

**Galvanic Skin Response** GSR is another name for Electrodermal activity (EDA) is the property of the human body that causes continuous variation in the electrical characteristics of the skin. See also: https://en.wikipedia.org/wiki/Electrodermal_activity. 73, 166

**generative theory of tonal music** a theory of music conceived by American composer and music theorist Fred Lerdahl and American linguist Ray Jackendoff and presented in the 1983 book of the same title. See also: https://en.wikipedia.org/wiki/Generative_theory_of_tonal_music Hamanaka et al., 2014. 42, 166

**Gesture Description Interchange Format** a format for storing, retrieving and sharing information about music-related gestures. See also: **Jensenius2006a**. 76, 166

**gigabyte** a unit of information equal to one thousand million (109) or, strictly, 230 bytes. See also: https://en.wikipedia.org/wiki/Gigabyte. 47, 166

**Graphical User Interface** The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. See also: https://en.wikipedia.org/wiki/Graphical_user_interface. 36, 166

**Graphics Processing Unit** a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. See also: https://en.wikipedia.org/wiki/Graphics_processing_unit. 48, 166

**GUIDO Music Notation Format** The GUIDO Music Notation Format is a formal language for score level music representation. It is a plain-text, i.e. readable and platform independent format capable of representing all information contained in conventional musical scores. See also: http://guidolib.sourceforge.net/GUIDO/. 57, 166

**Hewlett Packard** an American multinational information technology company headquartered in Palo Alto, California. See also: https://en.wikipedia.org/wiki/Hewlett-Packard. 38, 166

**Hierarchical Music Specification Language** HMSL is a programming language for experimental music composition and performance. It was popular between 1986 to 1996. HMSL is an object oriented set of extensions to the Forth language for the Amiga and the Macintosh. See also: http://www.softsynth.com/hmsl/. 62, 166

**High Performance Analytics Appliance** an in-memory, column-oriented, relational database management system developed and marketed by SAP SE See also: https://en.wikipedia.org/wiki/SAP_HANA. 38, 166

**Human Computer Interaction** A field that researches the design and use of computer technology, focused on the interfaces between people (users) and computers. See also: https://en.wikipedia.org/wiki/Human-computer_interaction. 52, 166

**Human Metabolome Database** a comprehensive, high-quality, freely accessible, online database of small molecule metabolites found in the human body. Created by the Human Metabolome Project funded by Genome Canada. See also: https://en.wikipedia.org/wiki/Human_Metabolome_Database. 47, 166

**Hypertext Markup Language** the standard markup language for creating web pages and web applications. With Cascading Style Sheets and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web See also: https://en.wikipedia.org/wiki/HTML. 166

**Hypertext Transfer Protocol** an application protocol for distributed, collaborative, hypermedia information systems. See also: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol. 48, 166

**I Operating System** a mobile operating system created and developed by Apple Inc. exclusively for its hardware. See also: https://en.wikipedia.org/wiki/IOS. 74, 166

**IBM-7090** The IBM 7090 is a second-generation transistorized version of the earlier IBM 709 vacuum tube mainframe computer that was designed for 'large-scale scientific and technological applications.' The first 7090 installation was in November 1959.. 138, 139, 166

**ICloud** cloud storage and cloud computing service from Apple Inc. launched on October 12, 2011. As of February 2016, the service had 782 million users.. 166

**In Memory Data Bases** a database management system that primarily relies on main memory for computer data storage. It is contrasted with database management systems that employ a disk storage mechanism. See also: https://en.wikipedia.org/wiki/In-memory_database. 166

**Incorporated Research Institutions for Seismology** IRIS is a consortium of over 120 US universities dedicated to the operation of science facilities for the acquisition, management, and distribution of seismological data. See also: https://www.iris.edu. 48, 166

**Information Management System** the first database model ever created. It was created by IBM during the early 1960s, in conjunction with two other American manufacturing conglomerates (Rockwell and Caterpillar) for NASA's Project Apollo See also: https://en.wikipedia.org/wiki/IBM_Information_Management_System. 30, 166

**Information Retrieval** the activity of obtaining information system resources relevant to an information need from a collection of information resources. Searches can be based on full-text or other content-based indexing. See also: https://en.wikipedia.org/wiki/Information_retrieval https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. 40, 166

**Institut de Recherche et Coordination Acoustique/Musique** a French institute for science about music and sound and avant garde electro-acoustical art music. See also: https://www.ircam.fr/. 61, 166

**Institute of Electronic Music and Acoustics** a multidisciplinary research center within the University of Music and Performing Arts, Graz, (Austria). See also: https://en.wikipedia.org/wiki/Institute_of_Electronic_Music_and_Acoustics. 45, 166

**Integrated Data Store** an early network database management system largely used by industry, known for its high performance. IDS became the basis for the CODASYL Data Base Task Group standards. See also: https://en.wikipedia.org/wiki/Integrated_Data_Store. 32, 166

**Integrated Database Management System** a network model database management system for mainframes See also: https://en.wikipedia.org/wiki/IDMS. 38, 166

**International Business Machines Corporation** an American multinational information technology company headquartered in Armonk, New York, with operations in over 170 countries. See also: https://en.wikipedia.org/wiki/IBM. 30, 166

**International Computer Music Conference** a yearly international conference for computer music researchers and composers. It is the annual conference of the International Computer Music Association (ICMA). See also: https://en.wikipedia.org/wiki/International_Computer_Music_Conference. 47, 166

**International Society for Music Information Retrieval** an international forum for research on the organization of music-related data. See also: http://ismir.net. 40, 166

**Internet Protocol** the principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet. See also: https://en.wikipedia.org/wiki/Internet_Protocol. 26, 166

**JavaScript Object Notation** an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value) See also: https://www.json.org/. 33, 166

**Linear Predictive Coding** a tool used mostly in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a linear predictive model See also: https://en.wikipedia.org/wiki/Linear_predictive_coding. 71, 166

**LISt Processor** a family of computer programming languages with a long history and a distinctive, fully parenthesized prefix notation. Originally specified in 1958, Lisp is the second-oldest high-level programming language in widespread use today. Linked lists are one of Lisp's major data structures, and Lisp source code is made of lists. See also: https://en.wikipedia.org/wiki/Lisp_(programming_language). 38, 166

**Local Boundaries Detection Model** The Local Boundary Detection Model (LBDM) calculates boundary strength values for each interval of a melodic surface according to the strength of local discontinuities; peaks in the resulting sequence of boundary strengths are taken to be potential local boundaries. See also: **DBLP:conf/icmc/Cambouropoulos01**. 166

**Looperman.com** Looperman is a Free pro audio community for musicians, film and video producers, djs and multi media designers. See also: https://www.looperman.com/. 41, 166

**madBPM** A modular C++ software platform serving as a data-ingestion engine suitable for database perceptualization (i.e. sonification and visualization). 50, 166

**Massachusetts Institute of Technology** private research university in Cambridge, Massachusetts. See also: http://www.mit.edu/. 41, 166

**Max** Max programming language. Named after Max Mathews.. 61–64, 144, 146, 166

**MAX/MSP** also known as Max/MSP/Jitter, is a visual programming language for music and multimedia developed and maintained by San Francisco-based software company Cycling '74. Max (Mathews) + ("Max Signal Processing", or the initials Miller Smith Puckette). 34, 49, 63, 66, 69, 71, 152, 166

**MetriX in Extensible Markup Language** A computer music synthesis programming language based in the Music-N type. See also: https://xamat.github.io/Thesis/html-thesis/ Amatriain, 2004. 76, 166

**Microsoft Access** Microsoft Access is a database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. See also: https://products.office.com/en/access. 38, 166

**MongoDB** a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata.. 26, 38, 166

**MuseData** The MuseData database is a project of the Center for Computer Assisted Research in the Humanities (CCARH). The database was created by Walter Hewlett. Data entry has been primarily done by Frances Bennion, Edmund Correia, Walter Hewlett, and Steve Rasmussen.. 41, 166

**Music Analysis, Retrieval and Synthesis for Audio Signals** Marsyas is an open source software framework for audio processing with specific emphasis on Music Information Retrieval applications. It has been designed and written by George Tzanetakis with help from students and researchers from around the world. Marsyas has been used for a variety of projects in both academia and industry. See also: http://marsyas.info/. 41, 166

**Music information retrieval** the interdisciplinary science of retrieving information from music. MIR is a small but growing field of research with many real-world applications. Those involved in MIR may have a background in musicology, psychoacoustics, psychology, academic music study, signal processing, informatics, machine learning, optical music recognition, computational intelligence or some combination of these. See also: https://en.wikipedia.org/wiki/Music_information_retrieval. 2, 166

**MUsic SImulator-Interpreter for COMpositional Procedures** "Development of MUSICOMP began in the late 1950s, and is considered by Loy as 'the granddaddy of all programming systems for automatic music generation' (1989, p. 368) and by Roads as the 'first composition language' (1996, p. 815).... From 1967 to 1969 these tools were used [with John Cage] in the production of HPSCHD" See also: (Ariza, 2005a, p. 44). 138, 166

**MUSIC-11** Barry Vercoe's development of MUSIC-IV which later grew into the still widely used Csound.. 58, 60, 166

**MUSIC-IV** See MUSIC-N. 166

**MUSIC-N** MUSIC-N refers to a family of computer music programs and programming languages descended from or influenced by MUSIC, a program written by Max Mathews in 1957 at Bell Labs.. 58, 166

**MUSIC-V** See MUSIC-N. 58, 63, 142–144, 166

**Music21** Music21 is a set of tools for helping scholars and other active listeners answer questions about music quickly and simply. It is a python module.. 41, 166

**Musical Instrument Digital Interface** a technical standard that describes a communications protocol, digital interface, and electrical connectors that connect a wide variety of electronic musical instruments, computers, and related audio devices. See also: https://en.wikipedia.org/wiki/MIDI. 55, 166

**Musical Instrument Museums Online** The world's largest freely accessible database for information on musical instruments held in public collections. See also: http://www.mimo-international.com. 41, 166

**MusicXML** MusicXML is an XML-based file format for representing Western musical notation. The format is open, fully documented, and can be freely used. See also: https://en.wikipedia.org/wiki/MusicXML. 57, 166

**MySQL** an open source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.. 26, 38, 73, 166

**National Aeronautics and Space Administration** an independent agency of the United States Federal Government responsible for the civilian space program, as well as aeronautics and aerospace research.NASA was established in 1958, succeeding the National Advisory Committee for Aeronautics (NACA). See also: https://www.nasa.gov/. 30, 166

**Neo4j** a graph database management system developed by Neo4j, Inc. . 38, 166

**net.loadbang-SQL** A Java library for communicating with SQL databases from MXJ. We currently support MySQL and HSQLDB. The HSQLDB system includes an embedded database instance, so it runs automatically from text files in Maxś search path; no external database server configuration is necessary.. 166

**NetworkX** NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. See also: http://networkx.github.io. 38, 166

**New Interfaces for Musical Expression** an international conference dedicated to scientific research on the development of new technologies and their role in musical expression and artistic performance. Researchers and musicians from all over the world gather to share their knowledge and late-breaking work on new musical interface design. See also: https://en.wikipedia.org/wiki/New_Interfaces_for_Musical_Expression. 47, 166

**New York University** a private research university spread throughout the world. See also: . 166

**NeXT** an American computer and software company founded in 1985 by Apple Computer cofounder Steve Jobs.. 59, 166

**Non or Not only SQL** a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. See also: https://en.wikipedia.org/wiki/NoSQL. 26, 166

**Nuclear magnetic resonance** a physical phenomenon in which nuclei in a strong static magnetic field are perturbed by a weak oscillating magnetic field (in the near field and therefore not involving electromagnetic waves) and respond by producing an electromagnetic signal with a frequency characteristic of the magnetic field at the nucleus. See also: https://en.wikipedia.org/wiki/Nuclear_magnetic_resonance. 47, 166

**Objective-C** Objective-C is a programming language combining the Smalltalk messaging system and the C programming language, which enables an object-oriented approach to the latter.. 59, 166

**OMax** OMax is a software environment (Creative Agent) which learns in real-time typical features of a musician's style and plays along with him interactively, giving the flavor of a machine co-improvisation. . 166

**Open Sound Control** a protocol for networking sound synthesizers, computers, and other multimedia devices for purposes such as musical performance or show control. OSC's advantages include interoperability, accuracy, flexibility and enhanced organization and documentation. See also: https://en.wikipedia.org/wiki/Open_Sound_Control. 49, 166

**OpenFrameworks** openFrameworks is an open source toolkit designed for creative coding. It is written in C++ and built on top of OpenGL. It runs on Microsoft Windows, macOS, Linux, iOS, Android and Emscripten. See also: https://openframeworks.cc. 50, 166

**Oracle Corporation** an American multinational computer technology corporation headquartered in Redwood Shores, California. See also: https://en.wikipedia.org/wiki/Oracle_Corporation. 38, 166

**Phase Vocoder** a collection of phase vocoder signal processing routines and accompanying shell scripts for use in the transformation and manipulation of sounds. It is written in C and designed to be used in a UNIX environment. See also: http://www.cs.princeton.edu/courses/archive/spr99/cs325/koonce.html. 166

**PostgreSQL** an open source object-relational database management system with an emphasis on extensibility and standards compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.. 26, 38, 75, 166

**QCD-Audio** QCD-audio works on data of computer physics, stemming from the Institute for Physics. Our interdisciplinary proposal QCD-Audio will develop sonification techniques for data of numerical models in physics.. 45, 166

**Quantuum Electrodynamics** In particle physics, quantum electrodynamics is the relativistic quantum field theory of electrodynamics See also: https://en.wikipedia.org/wiki/Quantum_electrodynamics. 46, 166

**Raima Database Manager** an ACID-compliant embedded database management system designed for use in embedded systems applications. RDM has been designed to utilize multi-core computers, networking (local or wide area), and on-disk or in-memory storage management. See also: https://en.wikipedia.org/wiki/Raima_Database_Manager. 38, 166

**Random Access Memory** a form of computer data storage that stores data and machine code currently being used. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. See also: https://en.wikipedia.org/wiki/Random-access_memory. 166

**Real World Computing Music Database** a copyright-cleared music database (DB) that is available to researchers as a common foundation for research. It was built by the RWC Music Database Sub-Working Group of the Real World Computing Partnership (RWCP) of Japan See also: https://staff.aist.go.jp/m.goto/RWC-MDB/. 166

**Real-Time Corpus-Based Concatenative Synthesis** The concatenative real-time sound synthesis system CataRT plays grains from a large corpus of segmented and descriptor-analysed sounds according to proximity to a target position in the descriptor space. This can be seen as a content-based extension to granular synthesis providing direct access to specific sound characteristics. See also: http://imtr.ircam.fr/imtr/CataRT. 69, 166

**Realtime Cmix** a real-time software "language" for doing digital sound synthesis and signal-processing. It is written in C/C++, and is distributed open-source, free of charge. See also: http://rtcmix.org/. 65, 166

**Redis** Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. 38, 166

**regions of interest** samples within a data set identified for a particular purpose. For example, in medical imaging, the boundaries of a tumor may be defined on an image or in a volume, for the purpose of measuring its size. See also: https://en.wikipedia.org/wiki/Region_of_interest. 45, 166

**Relational Model/Tasmania** Extensions to the Relational Model See also: https://en.wikipedia.org/wiki/Relational_Model/Tasmania. 166

**Repertoire International des Sources Musicales** The International Inventory of Musical Sources —Repertoire International des Sources Musicales (RISM)— is an international, non-profit organization which aims for comprehensive documentation of extant musical sources worldwide... Nearly all of the records may be downloaded as open data and linked open data in MARCXML and RDF format under a Creative Commons CC-BY license. See also: https://opac.rism.info. 41, 166

**SCORE** a scorewriter program, written in FORTRAN for DOS by Stanford Professor Leland Smith (1925-2013) with a reputation for producing very high-quality results.. 58, 60, 166

**SCRIVA** notation software for the SSSP. 52, 55, 166

**Sedna** Sedna is a free native XML database which provides a full range of core database services - persistent storage, ACID transactions, security, indices, hot backup. Flexible XML processing facilities include W3C XQuery implementation, tight integration of XQuery with full-text search facilities and a node-level update language.. 38, 166

**Short Time Fourier Transform** a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. See also: https://en.wikipedia.org/wiki/Short-time_Fourier_transform. 68, 166

**Sinusoidal Partial Editing Analysis and Resynthesis** an application for audio analysis, editing and synthesis. The analysis procedure (which is based on the traditional McAulay-Quatieri technique) attempts to represent a sound with many individual sinusoidal tracks (partials), each corresponding to a single sinusoidal wave with time varying frequency and amplitude. See also: http://www.klingbeil.com/spear/. 76, 166

**Smalltalk** Smalltalk is an object-oriented, dynamically typed reflective programming language.. 38, 59, 166

**SonART** a flexible, multi-purpose multimedia environment that allows for networked collaborative interaction with applications for art, science and industry. It provides an open ended framework for integration of powerful image and audio processing methods with a flexible network features.. 49, 166

**Sonifying Data** SonData is an Interactive Data Sonification toolkit, targeted at all practitioners interested in sonifying data. However, it also provides a set of tools that are useful to the academic and scientific Data Sonification community. See also: https://github.com/JoaoMenezes/SonData. 50, 166

**Sound Description Interchange Format** a standard for the well-defined and extensible interchange of a variety of sound descriptions. See also: https://en.wikipedia.org/wiki/SDIF. 76, 166

**Sparksee** Sparksee is a high-performance and scalable graph database management system written in C++. Its development started in 2006 and its first version was available on Q3 - 2008. The fourth version is available since Q3-2010.. 38, 166

**Sparsity** High-performance human solutions for Extreme Data. 38, 166

**Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrogram** a high-quality vocoder designed for speech analysis, modification, and synthesis See also: **icmc/bbp2372.1999.411**. 166

**SQLObject** SQLObject is a popular Object Relational Manager for providing an object interface to your database, with tables as classes, rows as instances, and columns as attributes. SQLObject includes a Python-object-based query language that makes SQL more abstract, and provides substantial database independence for applications.. 33, 38, 166

**Structured Query Language** a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system. See also: https://en.wikipedia.org/wiki/SQL. 33, 166

**Structured Query Language Lite** a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program. See also: https://en.wikipedia.org/wiki/SQLite. 38, 166

**Structured Sound Synthesis Project** an interdisciplinary project whose aim is to conduct research into problems and benefits arising from the use of computers in musical composition See also: https://www.billbuxton.com/SSSP.html. 28, 166

**Sybase** an enterprise software and services company that produced software to manage and analyze information in relational databases. . 166

**Synchronized Multimedia Integration Language** a World Wide Web Consortium recommended Extensible Markup Language markup language to describe multimedia presentations. It defines markup for timing, layout, animations, visual transitions, and media embedding, among other things. See also: https://en.wikipedia.org/wiki/Synchronized_Multimedia_Integration_Language. 166

**Telemeta** Telemeta is a free and open source collaborative multimedia asset management system (MAM) which introduces fast and secure methods to archive, backup, transcode, analyse, annotate and publish any digitalized video or audio file with extensive metadata. It is dedicated to collaborative media archiving projects, research laboratories and digital humanities — especially in ethno-musicological use cases — who need to easily organize and publish documented sound collections of audio files, CDs, digitalized vinyls and magnetic tapes over a strong database, through a smart and secure platform, in accordance with open web standards. Telemeta stands for Tele for “remote access” and meta for “metadata.”. 41, 166

**Timbre Identification** a collection of audio feature analysis externals for [Pd]. The classification extern (timbreID) accepts arbitrary lists of features and attempts to find the best match between an input feature and previously stored instances of training data. Besides doing identification, timbreID is also designed to facilitate real time concatenative synthesis and timbre-based orderings of sound sets. Its usage is fully explained in the accompanying helpfile. See also: http://williambrent.conflations.com/pages/research.html Brent, 2010. 166

**TimesTen** an in-memory, relational database management system with persistence and recoverability.. 166

**Transmission Control Protocol / Internet Protocol** The Internet protocol suite is the conceptual model and set of communications protocols used in the Internet and similar computer networks. It is commonly known as TCP/IP because the foundational protocols in the suite are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). See also: https://en.wikipedia.org/wiki/Internet_protocol_suite. 36, 166

**TurboIMAGE** a database developed by Hewlett Packard and included with the HP3000 minicomputer.. 38, 166

**UbuWeb** UbuWeb is a large web-based educational resource for avant-garde material available on the internet, founded in 1996 by poet Kenneth Goldsmith. It offers visual, concrete and sound poetry, expanding to include film and sound art mp3 archives.. 41, 166

**Uniform Resource Identifier** a string of characters that unambiguously identifies a particular resource. To guarantee uniformity, all URIs follow a predefined set of syntax rules, but also maintain extensibility through a separately defined hierarchical naming scheme. See also: https://en.wikipedia.org/wiki/Uniform_Resource_Identifier. 166

**Uniform Resource Locator** colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI) although many people use the two terms interchangeably. See also: https://en.wikipedia.org/wiki/URL. 166

**Unisys OS 2200 databases** The OS 2200 database managers are all part of the Universal Data System (UDS). UDS provides a common control structure for multiple different data models. Flat files (sequential, multi-keyed indexed sequential - MSAM, and fixed-block), network (DMS), and relational (RDMS) data models all share a common locking, recovery, and clustering mechanism. OS 2200 applications can use any mixtures of these data models along with the high-volume transaction file system within the same program while retaining a single common recovery mechanism. See also: https://en.wikipedia.org/wiki/Unisys_OS_2200_databases. 38, 166

**United Information Systems** Unisys Corporation is an American global information technology company based in Blue Bell, Pennsylvania, that provides a portfolio of IT services, software, and technology. It is the legacy proprietor of the Burroughs and UNIVAC line of computers, formed when the former bought the latter. See also: https://www.unisys.com. 38, 166

**Videoalive Indexer Extracted Closed Captions and Metadata** This is an index file in text form that shows closed captions and file offset information from a MPEG or WM video file. The .IXD file is used by VBrick Systems, Videoalive, Discovervideo.com, and other vendors. See also: https://filext.com/file-extension/IXD. 76, 166

**VMWare** a subsidiary of Dell Technologies that provides cloud computing and platform virtualization software and services.. 166

**WebDNA** a server-side scripting, interpreted language with an embedded database system, specifically designed for the World Wide Web. Its primary use is in creating database-driven dynamic web page applications.. 166

**XPlain** A semantic database model developed by J.H. ter Bekke. 38, 166

**YAML Ain't Markup Language** a human-readable data serialization language. It is commonly used for configuration files, but could be used in many applications where data is being stored or transmitted. See also: https://en.wikipedia.org/wiki/YAML. 33, 166

# Acronyms

**Access** Microsoft Access. 38, 166, *Glossary:* Microsoft Access

**AFLOWLIB** Automatic Flow for Materials Discovery. 49, 166, *Glossary:* Automatic Flow for Materials Discovery

**API** Application Programming Interface. 74, 166, *Glossary:* Application Programming Interface

**ASE** Adaptive Server Enterprise. 38, 166, *Glossary:* Adaptive Server Enterprise

**ASF** Apache Software Foundation. 166, *Glossary:* Apache Software Foundation

**AT** Actor-Network Theory. 166

**AutoCAD** Automatic Computer Assisted Design. 133, 166, *Glossary:* Automatic Computer Assisted Design

**BFCC** Bark Frequency Cepstrum Coefficient. 166

**BOOST** Boost Software Library. 38, 166, *Glossary:* Boost Software Library

**CAAC** Computer-Aided Algorithmic Composition. 29, 166, *Glossary:* Computer-Aided Algorithmic Composition

**CAC** Computer Aided Composition. 24, 59, 125, 128, 129, 139, 143, 166

**CAD** Computer Aided Design. 166, *Glossary:* Computer Aided Design

**CADDC** Computer Aided Data Driven Composition. 49, 166

**CAMP** Computer Assisted Music Project. 28, 55, 166, *Glossary:* Computer Assisted Music Project

**CASE** Computer Aided Software Engineering. 166, *Glossary:* Computer Aided Software Engineering

**CataRT** Real-Time Corpus-Based Concatenative Synthesis. 69, 166, *Glossary:* Real-Time Corpus-Based Concatenative Synthesis

**CC** Creative Commons. 77, 166, *Glossary:* Creative Commons

**CCRMA** Center for Computer Research in Music and Acoustics. 59, 71, 166, *Glossary:* Center for Computer Research in Music and Acoustics

**CDIP** Coastal Data Information Program. 47, 166, *Glossary:* Coastal Data Information Program

**CERL** Computer Based Education Research Laboratory. 62, 166, *Glossary:* Computer Based Education Research Laboratory

**CMAM** Centre des Musiques Arabes et Mediterraneennes. 41, 166, *Glossary:* Centre des Musiques Arabes et Mediterraneennes

**CNMAT** Center for New Music and Audio Technologies. 75, 166, *Glossary:* Center for New Music and Audio Technologies

**COBOL** Common Business Oriented Language. 31, 166, *Glossary:* Common Business Oriented Language

**CODASYL** Conference/Committee on Data Systems Languages. 31, 38, 166, *Glossary:* Conference/Committee on Data Systems Languages

**COMPath** Composition Path. 166, *Glossary:* Composition Path

**CouchDB** Appache Couch Database. 26, 38, 166, *Glossary:* Appache Couch Database

**CPU** central processing unit. 48, 66, 166, *Glossary:* central processing unit

**CSV** Comma Separated Values. 50, 166, *Glossary:* Comma Separated Values

**CUIDADO** Content Based Unified Interfaces and Descriptors for Audio/music Databases available Online. 70, 166, *Glossary:* Content Based Unified Interfaces and Descriptors for Audio/music Databases available Online

**CWM** Cluster Weighted Modeling. 166, *Glossary:* Cluster Weighted Modeling

**DARMS** Digital Alternate Representation of Musical Scores. 57–59, 166, *Glossary:* Digital Alternate Representation of Musical Scores

**DBMS** Database Managemet System. 24, 26, 28, 29, 75, 166, *Glossary:* Database Managemet System

**DDL** Data Definition Language. 29, 166, *Glossary:* Data Definition Language

**DJ** Disk Jockey. 70, 166, *Glossary:* Disk Jockey

**DML** data manipulation language. 29, 166, *Glossary:* data manipulation language

**DOM** Document Object Model. 32, 166, *Glossary:* Document Object Model

**DSL** Domain Specific Language. 50, 166, *Glossary:* Domain Specific Language

**EDM** Electronic dance music. 77, 166, *Glossary:* Electronic dance music

**EMI** Experiments in Music Intelligence. 128, 166, *Glossary:* Experiments in Music Intelligence

**ER** Entity Relationship. 166, *Glossary:* Entity Relationship

**EsAC** Essen Associative Code. 41, 166, *Glossary:* Essen Associative Code

**FDN** Feedback Delay Network. 84, 166, *Glossary:* Feedback Delay Network

**FM** Frequency Modulation. 48, 166, *Glossary:* Frequency Modulation

**FORTRAN** Formula Translation. 166, *Glossary:* Formula Translation

**GB** gigabyte. 47, 166, *Glossary:* gigabyte

**GDIF** Gesture Description Interchange Format. 76, 166, *Glossary:* Gesture Description Interchange Format

**GPU** Graphics Processing Unit. 48, 166, *Glossary:* Graphics Processing Unit

**GSR** Galvanic Skin Response. 73, 166, *Glossary:* Galvanic Skin Response

**GTTM** generative theory of tonal music. 42, 166, *Glossary:* generative theory of tonal music

**GUI** Graphical User Interface. 36, 52, 55, 166, *Glossary:* Graphical User Interface

**GUIDO** GUIDO Music Notation Format. 57, 166, *Glossary:* GUIDO Music Notation Format

**HCI** Human Computer Interaction. 52, 53, 166, *Glossary:* Human Computer Interaction

**HMDB** Human Metabolome Database. 47, 166, *Glossary:* Human Metabolome Database

**HMSL** Hierarchical Music Specification Language. 62, 63, 166, *Glossary:* Hierarchical Music Specification Language

**HP** Hewlett Packard. 38, 166, *Glossary:* Hewlett Packard

**HTML** Hypertext Markup Language. 166, *Glossary:* Hypertext Markup Language

**HTTP** Hypertext Transfer Protocol . 48, 166, *Glossary:* Hypertext Transfer Protocol

**IBM** International Business Machines Corporation. 30, 38, 166, *Glossary:* International Business Machines Corporation

**ICMC** International Computer Music Conference. 47, 58, 60, 64, 65, 166, *Glossary:* International Computer Music Conference

**IDMS** Integrated Database Management System. 38, 166, *Glossary:* Integrated Database Management System

**IDS** Integrated Data Store. 32, 38, 166, *Glossary:* Integrated Data Store

**IEM** Institute of Electronic Music and Acoustics. 45, 166, *Glossary:* Institute of Electronic Music and Acoustics

**IFT** Inverse Fourier Transform. 47, 49, 166

**IMDB** Internet Movie Database. 35, 166

**IMDBs** In Memory Data Bases. 166, *Glossary:* In Memory Data Bases

**IMS** Information Management System. 30, 38, 166, *Glossary:* Information Management System

**iOS** I Operating System. 74, 166, *Glossary:* I Operating System

**IP** Internet Protocol. 26, 48, 166, *Glossary:* Internet Protocol

**IR** Information Retrieval. 40, 166, *Glossary:* Information Retrieval

**IRCAM** Institut de Recherche et Coordination Acoustique/Musique. 61, 65, 68, 70, 131, 132, 166, *Glossary:* Institut de Recherche et Coordination Acoustique/Musique

**IRIS** Incorporated Research Institutions for Seismology. 48, 166, *Glossary:* Incorporated Research Institutions for Seismology

**ISMIR** International Society for Music Information Retrieval. 40, 166, *Glossary:* International Society for Music Information Retrieval

**IXD** Videoalive Indexer Extracted Closed Captions and Metadata. 76, 166, *Glossary:* Videoalive Indexer Extracted Closed Captions and Metadata

**JSON** JavaScript Object Notation. 33, 35, 76, 166, *Glossary:* JavaScript Object Notation

**LBDM** Local Boundaries Detection Model. 166, *Glossary:* Local Boundaries Detection Model

**LFCC** Log Frequency Cepstral Coefficients. 70, 166

**LISP** LISt Processor. 38, 65, 128, 166, *Glossary:* LISt Processor

**Looperman** Looperman.com. 41, 70, 166, *Glossary:* Looperman.com

**LPC** Linear Predictive Coding. 71, 166, *Glossary:* Linear Predictive Coding

**Marsyas** Music Analysis, Retrieval and Synthesis for Audio Signals. 41, 166, *Glossary:* Music Analysis, Retrieval and Synthesis for Audio Signals

**MetriXML** MetriX in Extensible Markup Language. 76, 166, *Glossary:* MetriX in Extensible Markup Language

**MIDI** Musical Instrument Digital Interface. 55, 57, 61, 66, 128, 166, *Glossary:* Musical Instrument Digital Interface

**MIMO** Musical Instrument Museums Online. 41, 166, *Glossary:* Musical Instrument Museums Online

**MIR** Music information retrieval . 2, 39–41, 70, 78, 166, *Glossary:* Music information retrieval

**MIT** Massachusetts Institute of Technology . 41, 52, 61, 64, 166, *Glossary:* Massachusetts Institute of Technology

**MUSICOMP** MUsic SImulator-Interpreter for COMpositional Procedures. 138, 166, *Glossary:* MUsic SImulator-Interpreter for COMpositional Procedures

**MusicXML** MusicXML. 57, 166, *Glossary:* MusicXML

**NASA** National Aeronautics and Space Administration. 30, 166, *Glossary:* National Aeronautics and Space Administration

**NC2IF** Center for Network-Centric Cognition and Information Fusion. 48, 166, *Glossary:* Center for Network-Centric Cognition and Information Fusion

**NetworkX** NetworkX. 38, 166, *Glossary:* NetworkX

**NIME** New Interfaces for Musical Expression. 47, 166, *Glossary:* New Interfaces for Musical Expression

**NMR** Nuclear magnetic resonance. 47, 166, *Glossary:* Nuclear magnetic resonance

**NoSQL** Non or Not only SQL. 26, 33, 38, 166, *Glossary:* Non or Not only SQL

**NYU** New York University. 166, *Glossary:* New York University

**OFX** OpenFrameworks. 50, 166, *Glossary:* OpenFrameworks

**Oracle** Oracle Corporation. 38, 166, *Glossary:* Oracle Corporation

**OS 2200** Unisys OS 2200 databases. 38, 166, *Glossary:* Unisys OS 2200 databases

**OSC** Open Sound Control . 49, 75, 166, *Glossary:* Open Sound Control

**PVC** Phase Vocoder. 166, *Glossary:* Phase Vocoder

**QED** Quantuum Electrodynamics. 46, 166, *Glossary:* Quantuum Electrodynamics

**RAM** Random Access Memory . 166, *Glossary:* Random Access Memory

**RDM** Raima Database Manager . 38, 166, *Glossary:* Raima Database Manager

**RISM** Repertoire International des Sources Musicales. 41, 166, *Glossary:* Repertoire International des Sources Musicales

**RM/T** Relational Model/Tasmania. 166, *Glossary:* Relational Model/Tasmania

**ROI** regions of interest. 45, 166, *Glossary:* regions of interest

**RTcmix** Realtime Cmix. 65–67, 166, *Glossary:* Realtime Cmix

**RWC** Real World Computing Music Database. 166, *Glossary:* Real World Computing Music Database

**SAP HANA** High Performance Analytics Appliance. 38, 166, *Glossary:* High Performance Analytics Appliance

**SDIF** Sound Description Interchange Format. 76, 166, *Glossary:* Sound Description Interchange Format

**SMC** Sound and Music Computing Conference. 47, 166

**SMIL** Synchronized Multimedia Integration Language. 166, *Glossary:* Synchronized Multimedia Integration Language

**SonData** Sonifying Data. 50, 166, *Glossary:* Sonifying Data

**SPEAR** Sinusoidal Partial Editing Analysis and Resynthesis. 76, 166, *Glossary:* Sinusoidal Partial Editing Analysis and Resynthesis

**SQL** Structured Query Language. 33, 71, 166, *Glossary:* Structured Query Language

**SQLite** Structured Query Language Lite. 38, 166, *Glossary:* Structured Query Language Lite

**SSSP** Structured Sound Synthesis Project. 28, 52, 53, 55, 59, 64, 166, *Glossary:* Structured Sound Synthesis Project

**STFT** Short Time Fourier Transform. 68, 166, *Glossary:* Short Time Fourier Transform

**STRAIGHT** Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrogram. 166, *Glossary:* Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrogram

**TCP/IP** Transmission Control Protocol / Internet Protocol. 36, 66, 166, *Glossary:* Transmission Control Protocol / Internet Protocol

**timbreID** Timbre Identification. 166, *Glossary:* Timbre Identification

**Unisys** United Information Systems. 38, 166, *Glossary:* United Information Systems

**URI** Uniform Resource Identifier. 166, *Glossary:* Uniform Resource Identifier

**URL** Uniform Resource Locator . 166, *Glossary:* Uniform Resource Locator

**XML** Extensible Markup Language. 33, 35, 38, 76, 166, *Glossary:* Extensible Markup Language

**YAML** YAML Ain't Markup Language. 33, 35, 76, 166, *Glossary:* YAML Ain't Markup Language

# Bibliography


Abiteboul, S. (1996). *Querying semi-structured data* (Technical Report No. 1996-19). Stanford InfoLab. Stanford InfoLab. Retrieved from http://ilpubs.stanford.edu:8090/144/

Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of databases*. Addison-Wesley.

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

Ames, C. (1985). Applications of linked data structures to automated composition. In *Proceedings of the international computer music conference, ICMC 1985*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.1985.040/1

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, *40*(1).

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. (2003). Ornament as data structure: An algorithmic model based on micro-rhythms of csng laments and funeral music. In *Proceedings of the international computer music conference, ICMC 2003*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2003.030/1

Ariza, C. (2005a). *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

Ariza, C. (2005b). The xenakis sieve as object: A new model and a complete implementation. *Computer Music Journal*, *29*(2), 40–60. Retrieved from https://www.jstor.org/stable/3681712

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

Assayag, G., Dubnov, S., & Delerue, O. (1999). Guessing the composer's mind: Applying universal prediction to musical style. In *Proceedings of the 1999 international computer music conference, ICMC 1999, beijing, china, october 22-27, 1999*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1999.454

Attali, J. (2009). *Noise: The political economy of music*. University of Minnesota Press.

Bachman, C. W. (1973). The programmer as navigator. *Commun. ACM*, *16*(11), 653–658. doi:10.1145/355611.362534

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

Barthes, R., Lavers, A., & Smith, C. (1968). *Elements of semiology*. Hill and Wang, New York.

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

Bloch, G., & Dubnov, S. (2008). Introducing video features and spectral descriptors in the omax improvisation system. In *Proceedings of the 2008 international computer music conference, ICMC 2008, belfast, ireland, august 24-29, 2008*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2008.090

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 informa-*

*tion 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

Boie, R., Mathews, M., & Schloss, A. (1989). The radio drum as a synthesizer controller. In *Proceedings of the 1989 international computer music conference, ICMC 1989, columbus, ohio, usa, november 2-5, 1989*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.1989.010

Borges, J. L. (1942). Funes el memorioso. *Ficciones*.

Born, G. (1995). *Rationalizing culture*. University of California Press.

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

Brent, W. (2010). A timbre analysis and classification toolkit for pure data. In *Proceedings of the international computer music conference, ICMC 2010*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2010.044/1

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

Bullock, J., & Frisk, H. (2009). An object oriented model for the representation of temporal data in the integra framework. In *Proceedings of the international computer music conference, ICMC 2009*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2009.012/1

Buneman, P. (1997). Semistructured data. In *Proceedings of the sixteenth acm sigact-sigmod-sigart symposium on principles of database systems* (pp. 117–121). PODS '97. doi:10.1145/263661.263675

Butler, J. (1988). Performative acts and gender constitution: An essay in phenomenology and feminist theory. *Theatre Journal*, *40*(4).

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

Cámara Halac, F. (2018b). This is for young ears: A response to elsa justel's marelle... *Open Space*, (21), 339–350.

Caramiaux, B., Bevilacqua, F., & Schnell, N. (2011). Sound selection by gestures. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 329–330). Oslo, Norway. Retrieved from http://www.nime.org/proceedings/2011/nime2011_329.pdf

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/

Carpentier, G., Tardieu, D., Rodet, X., & Saint-James, E. (2006). Imitative and Generative Orchestrations Using Pre- analysed Sounds Databases. doi:10.5281/zenodo.849343

Carter, R. (2017). On the expressive potential of suboptimal speakers. audience participation on mobile devices.

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

Cartwright, M., & Pardo, B. (2014). Synthassist: Querying an audio synthesizer by vocal imitation. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 363–366). London, United Kingdom: Goldsmiths, University of London. Retrieved from http://www.nime.org/proceedings/2014/nime2014_446.pdf

Cascone, K. (2000). The aesthetics of failure: 'post-digital' tendencies in contemporary computer music. *Computer Music Journal*, *24*(4), 12–18. Retrieved from https://doi.org/10.1162/014892600559489

Casey, M. A., & Grierson, M. (2007). Soundspotter / remix-tv: Fast approximate matching for audio and video performance. In *Proceedings of the 2007 international computer music conference, ICMC 2007, copenhagen, denmark, august 27-31, 2007*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2007.205

Casey, M. A., & Slaney, M. (2006). Song intersection by approximate nearest neighbor search. In *ISMIR 2006, 7th international conference on music information retrieval, victoria, canada, 8-12 october 2006, proceedings* (pp. 144–149).

Choi, I. (2000). Voices in ruins — composition with residuals. Retrieved from https://vimeo.com/23086026

Choi, I., Zheng, G., & Chen, K. (2000). Embedding a sensory data retrieval system in a movement-sensitive space and a surround sound system. In *Proceedings of the international computer music conference, ICMC 2000*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2000.146/1

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

Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM*, *13*(6), 377–387. doi:10.1145/362384.362685

Codd, E. F. (1972). Relational completeness of data base sublanguages. In *Database systems* (pp. 65–98). Prentice-Hall.

Collins, N. (2006). *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems* (Doctoral dissertation, University of Cambridge).

Collins, N. (2007). Audiovisual concatenative synthesis. In *Proceedings of the 2007 international computer music conference, ICMC 2007, copenhagen, denmark, august 27-31, 2007*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2007.187

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

Collins, N., Mclean, A., Rohrhuber, J., & Ward, A. (2003). Live coding in laptop performance. *Organised Sound*, *8*, 321–329. doi:10.1017/S135577180300030X

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

Cope, D. (1987a). An expert system for computer-assisted composition. *Computer Music Journal*, *11*(4), 30–46. Retrieved from http://www.jstor.org/stable/3680238

Cope, D. (1987b). Experiments in music intelligence (EMI). In *ICMC*, Michigan Publishing.

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

Crowley, C. (1998). Data structures for text sequences. In . Retrieved from https://www.cs.unm.edu/~crowley/papers/sds.pdf

Daniel, S. (2007). The database: An aesthetics of dignity. *Database Aesthetics: Art in the Age of Information Overflow*.

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

Derrida, J. (1978). *Writing and difference*. The University of Chicago.

Derrida, J. (1982). *Margins of philosophy*. The Harvester Press.

Derrida, J., & Prenowitz, E. (1995). Archive fever: A freudian impression. *Diacritics*, *25*(2).

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, oc-*

*tober 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

Eco, U. (2004). The poetics of the open work. *Audio Culture: Readings in Modern Music*.

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

Ernst, W. (2013). *Digital memory and the archive*. University of Minnesota Press.

Esposito, A., Krichevsky, R., & Nicolis, A. (2019). Gravitational mass carried by sound waves. *Phys. Rev. Lett. 122*, 084501. doi:10.1103/PhysRevLett.122.084501

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.

Frisson, C. (2015). *Designing interaction for browsing media collections (by similarity)* (Doctoral dissertation, Universit de Mons). Retrieved from http://tcts.fpms.ac.be/publications/phds/frisson/phd-frisson.pdf

García, F., Vinceslas, L., Tubau, J., & Maestre, E. (2011). Acquisition and study of blowing pressure profiles in recorder playing. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 124–127). Oslo, Norway. Retrieved from http://www.nime.org/proceedings/2011/nime2011_124.pdf

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

Gratton, P., & Morin, M.-E. (2015). *The nancy dictionary*. Edinburgh University Press.

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

Hamilton, R. (2006). Bioinformatic response data as a compositional driver. In *Proceedings of the international computer music conference, ICMC 2006*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2006.123/1

Hansen, M. B. N. (2002). Cinema beyond cybernetics, or how to frame the digital image. *Configurations*, *10*(1).

Hansen, M. B. N. (2004). *New philosophy for new media*. The MIT Press.

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

Hayles, N. K. (1993). The materiality of informatics. *Configurations*, *1*(1).

Hayles, N. K. (1999). *How we became posthuman: Virtual bodies in cybernetics, literature, and informatics*. The University of Chicago Press.

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.

Hiller, L. A., & Isaacson, L. M. (1959). *Experimental music: Composition with an electronic computer*. McGraw-Hill Book Company, Inc.

Hochenbaum, J., Kapur, A., & Wright, M. (2010). Multimodal musician recognition. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 233–237). Sydney, Australia. Retrieved from http://www.nime.org/proceedings/2010/nime2010_233.pdf

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

IV, J. A. M. (1999). *A brief history of algorithmic composition*. Online. Retrieved from https://ccrma.stanford.edu/~blackrse/algorithm.html

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 expression* (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

Jones, R., Lagrange, M., & Schloss, W. A. (2007). A hand drumming dataset for physical modeling. In *Proceedings of the 2007 international computer music conference, ICMC 2007, copenhagen, denmark, august 27-31, 2007*, Michigan Publishing. Retrieved from http://hdl.handle.net/2027/spo.bbp2372.2007.083

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

Kamde, P. M., & Algur, S. P. (2011). A survey on web multimedia mining. *CoRR*, *abs/1109.1145*. arXiv: 1109.1145. Retrieved from http://arxiv.org/abs/1109.1145

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

Kernighan, B. W. (1978). *The c programming language*. Englewood Cliffs, N.J.: Prentice-Hall.

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.

Klein, N. M. (2007). Waiting for the world to explode: How data convert into a novel. *Database Aesthetics: Art in the Age of Information Overflow*.

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

Kobayashi, R. (2003). Sound clustering synthesis using spectral data. In *Proceedings of the international computer music conference, ICMC 2003*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2003.052/1

Kroher, N. (2011). *Acoustic feedbacks in sound reinforcement systems: Investigating the larsen effect*. AV Akademikerverlag.

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.

Latour, B. (1990). On actor-network theory. a few clarifications plus more than a few complications. *Philosophia*, *25*(3).

Latour, B. (1993). *We have never been modern*. Harvard University Press Cambridge, Massachusetts.

Lewis, G. (2000). Too many notes: Computers, complexity, and culture in voyager. *Leonardo Music Journal*, *10*.

Lewis, G. E. (1999). Interacting with latter-day musical automata. *Contemporary Music Review*, *18*(3), 99–112. doi:10.1080/07494469900640381

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

Liu, Q., Han, Y. C., Kuchera-Morin, J., & Wright, M. (2013). Cloud bridge: A data-driven immersive audio-visual software interface. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 431–436). Daejeon, Republic of Korea: Graduate School of Culture Technology, KAIST. Retrieved from http://nime.org/proceedings/2013/nime2013_250.pdf

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

Long, R., Harrington, M., Hain, R., & Nicholls, G. (2000). *Ims primer*. International Business Machines Corporation. Retrieved from http://www.redbooks.ibm.com/redbooks/pdfs/sg245352.pdf

Loviscach, J. (2008). Programming a music synthesizer through data mining. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 221–224). Genoa, Italy. Retrieved from http://www.nime.org/proceedings/2008/nime2008_221.pdf

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

Lucier, A. (1970). I am sitting in a room. For voice and electronic tape. Retrieved from http://alucier.web.wesleyan.edu/discography.html

m zmölnig, I., & Eckel, G. (2015). *Live coding: An overview*.

Manovich, L. (2001). *The language of new media*. MIT Press.

Manovich, L. (2002). Old media as new media: Cinema. *The New Media Book*.

Mathews, M. V. (1963). The digital computer as a musical instrument. *Science*, *142*(3592), 553–557. Retrieved from http://www.jstor.org/stable/1712380

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

Mazzoni, D., & Dannenberg, R. B. (2001). A fast data structure for disk-based audio editing. In *Proceedings of the international computer music conference, ICMC 2001*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2001.051/1

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

Morin, P. (2019). *Open data structures*. Creative Commons. Retrieved from opendatastructures.org

Morton, T. (2013). *Hyperobjects: Philosophy and ecology after the end of the world*. University of Minnesota Press.

N. Shepard, R. (1964). Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, *36*, 2346. doi:10.1121/1.1919362

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

Nakamoto, M., & Kuhara, Y. (2007). Circle canon chorus system used to enjoy a musical ensemble singing "frog round". In *Proceedings of the international conference on new interfaces for musical expression* (pp. 409–410). New York City, NY, United States. Retrieved from http://www.nime.org/proceedings/2007/nime2007_409.pdf

Nancy, J.-L. (1991). *The inoperative community*. University of Minnesota Press, Minneapolis and Oxford.

Nancy, J.-L. (2007). *Listening*. Fordham University Place.

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

Nilson, C. (2007). Live coding practice. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 112–117). New York City, NY, United States. Retrieved from http://www.nime.org/proceedings/2007/nime2007_112.pdf

Nilson, C. (2016). *Collected rewritings: Live coding thoughts, 1968-2015.* Retrieved from https://composerprogrammer.com

Norman, A., & Amatriain, X. (2007). Data jockey, a tool for meta-data enhanced digital djing and active listening. In *Proceedings of the international computer music conference, ICMC 2007*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2007.117/1

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

Nuanàin, C. Ó., Jordà, S., & Herrera, P. (2016). An interactive software instrument for real-time rhythmic concatenative synthesis. In *Proceedings of the international conference on new interfaces for musical expression* (Vol. 16, pp. 383–387). 2220-4806. Brisbane, Australia: Queensland Conservatorium Griffith University. Retrieved from http://www.nime.org/proceedings/2016/nime2016_paper0076.pdf

Nymoen, K., & Jensenius, A. R. (2011). A Toolbox for Storing and Streaming Music-related Data. doi:10.5281/zenodo.849865

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

Oviedo, M. B. (2019). *Memoria, olvido y narración: Funes como antítesis del escritor*. Lecture given at Cornell University.

Paul, C. (2007). The database as system and cultural form: Anatomies of cultural narratives. *Database Aesthetics: Art in the Age of Information Overflow*.

Pauletto, S., & Hunt, A. (2004a). A toolkit for interactive sonification, Georgia Institute of Technology. Retrieved from http://hdl.handle.net/1853/50809

Pauletto, S., & Hunt, A. (2004b). 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.*

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

Poster, M. (2011). Introduction. *Into the Universe of Technical Images*.

Price, R., & Rebelo, P. (2008). Database and mapping design for audiovisual prepared radio set installation. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 311–314). Genoa, Italy. Retrieved from http://www.nime.org/proceedings/2008/nime2008_311.pdf

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. (2007). On timbre stamps and other frequency-domain filters. In *ICMC*, Michigan Publishing.

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

Roads, C. [C.], & Mathews, M. (1980). Interview with max mathews. *Computer Music Journal*, *4*(4), 15–22. Retrieved from http://www.jstor.org/stable/3679463

Roads, C. [Curtis]. (2001). *Microsound*. MIT Press.

Roberts, C., Wright, M., Kuchera-Morin, J., & Höllerer, T. (2014). Rapid creation and publication of digital musical instruments. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 239–242). London, United Kingdom: Goldsmiths, University of London. Retrieved from http://www.nime.org/proceedings/2014/nime2014_373.pdf

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

Rowe, R. (1992). *Interactive music systems: Machine listening and composing*. Cambridge, MA, USA: MIT Press. Retrieved from https://wp.nyu.edu/robert_rowe/text/interactive-music-systems-1993

Rowe, R., Garton, B., Desain, P., Honing, H., Dannenberg, R., Jacobs, D., ... Lewis, G. (1993). Editor's notes: Putting max in perspective. *Computer Music Journal*, *17*(2), 3–11. Retrieved from http://www.jstor.org/stable/3680864

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

Schloss, W., Driessen, A., Peter, & F. (2001). Towards a virtual membrane: New algorithms and technology for analyzing gestural data. In *Proceedings of the international computer music conference, ICMC 2001*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2001.103/1

Schmeder, A. (2009). Efficient gesture storage and retrieval for multiple applications using a relational data model of open sound control. In *Proceedings of the international computer music conference, ICMC 2009*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2009.005/1

Schöner, B., Cooper, C., Douglas, C., & Gershenfeld, N. (1998). Data-driven modeling and synthesis of acoustical instruments. 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.bbp2372.1998.265

Schwarz, D. (2000). A system for data-driven concatenative sound synthesis. In *Proceedings of the cost g-6 conference on digital audio effects (dafx-00), verona, italy, december 7-9*.

Schwarz, D. (2003). New developments in data-driven concatenative sound synthesis. In *Proceedings of the international computer music conference, ICMC 2003*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2003.099/1

Schwarz, D. (2006a). Concatenative sound synthesis: The early years. *Journal of New Music Research*, *35*, 3–22. doi:10.1080/09298210600696857

Schwarz, D. (2006b). Real-time corpus-based concatenative synthesis with catart. (pp. 18–21).
Schwarz, D. (2012). The sound space as musical instrument: Playing corpus-based concatenative synthesis. 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_120.pdf
Schwarz, D., & Schnell, N. (2009). Sound Search by Content-based Navigation in Large Databases. doi:10.5281/zenodo.849679
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
Serafin, S., Smith, J., III, O., Thornburg, H., Mazzella, F., Tellier, A., & Thonier, G. (2001). Data driven identification and computer animation of bowed string model. In *Proceedings of the international computer music conference, ICMC 2001*, Michigan Publishing. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2001.071/1
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
Shanks, D., & W.jun. Wrench, J. (1962). Calculation of pi to 100,000 decimals. *Mathematics of Computation*, *16*. doi:10.2307/2003813
Silberschatz, A., Stonebraker, M., & Ullman, J. (1995). *Database research: Achievements and opportunities into the 21st century* (Technical Report No. 1995-15). Stanford InfoLab. Stanford InfoLab. Retrieved from http://ilpubs.stanford.edu:8090/81/
Simonelli, L., Delgadino, M., & Cámara Halac, F. (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, october 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
Solomos, M. (2005). An introduction to horacio vaggione musical-theoretical thought. *Contemporary Music Review*, *25*(4), 311–326. Retrieved from https://hal.archives-ouvertes.fr/hal-00770212
Sterne, J. (2012). *Mp3: The meaning of a format*. Duke University Press.

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. (2004). Matconcat: An application for exploring concatenative sound synthesis using matlab.

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

Szendy, P. (2008). *Listen: A history of our ears*. Fordham University. Retrieved from https://www.jstor.org/stable/j.ctt13x002m

Taylor, B., Allison, J., Conlin, W., Oh, Y., & Holmes, D. (2014). Simplified expressive mobile development with nexusui, nexusup, and nexusdrop. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 257–262). London, United Kingdom: Goldsmiths, University of London. Retrieved from http://www.nime.org/proceedings/2014/nime2014_480.pdf

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.

Vaggione, H. (1993). Determinism and the false collective about models of time in early computer-aided composition. *Contemporary Music Review*, *7*(2).

Vaggione, H. (2001). Some ontological remarks about music composition processes. *Computer Music Journal*, *25*(1), 54–61.

Valle, A., & Sanfilippo, D. (2012). Towards a typology of feedback systems. In *Proceedings of the international computer music conference, ICMC 2012*. Retrieved from https://quod.lib.umich.edu/i/icmc/bbp2372.2012.006

van Eck, C. (2013). *Between air and electricity: Microphones and loudspeakers as musical instruments* (Doctoral dissertation, Leiden University).

Varese, E. (2004). The liberation of sound. *Audio Culture: Readings in Modern Music*.

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

Vesna, V. (2007a). *Database aesthetics: Art in the age of information overflow*. University of Minnesotta Press.

Vesna, V. (2007b). Seeing the world in a grain of sand: The database aesthetics of everything. *Database Aesthetics: Art in the Age of Information Overflow*.

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

Vinet, H. (2005). The semantic hifi project. 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.bbp2372.2005.171

Vinet, H., Herrera, P., & Pachet, F. (2002a). The CUIDADO project. In *ISMIR 2002, 3rd international conference on music information retrieval, paris, france, october 13-17, 2002, proceedings*. Retrieved from http://ismir2002.ismir.net/proceedings/02-FP06-3.pdf

Vinet, H., Herrera, P., & Pachet, F. (2002b). The CUIDADO project: New applications based on audio and music content description. 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.bbp2372.2002.093

Visi, F., Caramiaux, B., Mcloughlin, M., & Miranda, E. (2017). A knowledge-based, data-driven method for action-sound mapping. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 231–236). Copenhagen, Denmark: Aalborg University Copenhagen. Retrieved from http://www.nime.org/proceedings/2017/nime2017_paper0043.pdf

Vogl, R., & Knees, P. (2017). An intelligent drum machine for electronic dance music production and performance. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 251–256). Copenhagen, Denmark: Aalborg University Copenhagen. Retrieved from http://www.nime.org/proceedings/2017/nime2017_paper0047.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

von Neumann, J., & Burks, A. (1946). Preliminary discussion of the logical design of an electronic computing instrument. *Engineering, College of - Technical Reports*. Retrieved from https://library.ias.edu/files/Prelim_Disc_Logical_Design.pdf

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

Weinbren, G. (2007). Ocean, database, recut. *Database Aesthetics: Art in the Age of Information Overflow*.

Wessel, I., & Moulds, M. L. (2008). How many types of forgetting? comments on connerton (2008). *Memory Studies*, *1*(3).

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

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

Xambo, A., Roma, G., Lerch, A., Barthet, M., & Fazekas, G. (2018). Live repurposing of sounds: Mir explorations with personal and crowdsourced databases. In T. M. Luke Dahl Douglas Bowman (Ed.), *Proceedings of the international conference on new interfaces for musical expression* (pp. 364–369). Blacksburg, Virginia, USA: Virginia Tech. Retrieved from http://www.nime.org/proceedings/2018/nime2018_paper0081.pdf

Xenakis, I. (1992). *Formalized music: Thought and mathematics in music*. Pendragon Revised Edition.

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).

Young, D., & Deshmane, A. (2007). Bowstroke database : A web-accessible archive of violin bowing data. In *Proceedings of the international conference on new interfaces for musical expression* (pp. 352–357). New York City, NY, United States. Retrieved from http://www.nime.org/proceedings/2007/nime2007_352.pdf

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