

BUZU: Retrieving public transportation data for sonification

Julián Jaramillo Arango^{1*}, Esteban Viveros Astorga¹, Fernando Iazzetta¹, Cristiano Figueiró¹

¹Núcleo de Pesquisas em Sonologia – Escola de Comunicações e Artes, Universidade de São Paulo
Lúcio Martins Rodrigues, 443 - Butantã, São Paulo - SP, 05508-020, Brasil

julianjaus@yahoo.com, emviveros@gmail.com, iazzetta@usp.br, figocris@gmail.com

Abstract. *This paper discusses the conception, design and exhibition of BUZU, an audiovisual installation that generates an auditory image of the São Paulo bus transportation system. BUZU makes perceptible information of both the system’s planning and behavior during a particular week in October 2017. The work is an artistic outcome of the InterSCity project, an inter-institutional research initiative concerning the Future Internet and the Smart Cities. Along with the discussion of the BUZU creation process we will examine mining and processing strategies related to the sonification of big data, the data-to-sound mapping methods, the auditory structure for displaying the material and the public exhibition of the work in the context of an artistic event.*

1 Introduction

BUZU is an audiovisual installation risen as an artistic outcome of the InterSCity project [1], which designs computational strategies and tools for the Future Internet and the Smart Cities [2]. BUZU was launched in the “Sons de Silício” Art exhibition [3], that was held in the older São Paulo Museum of Contemporary Art in April 2019. BUZU emerged as the result of collaboration between the Research Center on Sonology (a.k.a NuSom) and the InterSCity.



Figure 1: The BUZU installation

The installation proposes an acoustical image of São Paulo by retrieving information of the city public bus transportation system, which comprises 2,183 lines. BUZU makes perceptible a dataset created by the InterSCity project, which reports the system’s behaviour in a particular working week in October, 2017. While the dataset original purpose is comparing the system behaviour with

*Supported by CAPES.

the Easter holiday, in BUZU the dataset is implemented to feed an audiovisual engine created in Pure Data (a.k.a. Pd) and Processing. The audio is projected by a quadraphonic system and the visuals are displayed in a central screen. Textual information is presented in four little LCD screens distributed around the central structure. By exploring alternative representations of the city, the project adopts sonification as the main strategy.

It is worth mentioning that sonification has been increasingly implemented in projects creating acoustical representations of the city. These projects involve the displaying of the London subway system information with reference sounds [4], the exploration of mapping techniques used in urban planning and design [5] and the organization of geospatial data in geographic maps [6]. An inspirational project is the metrosynth [7] which creates a sonification system of the Montreal subway system running in HTML5.

Instead, Buzu’ deals with a huge volume of data, which brought a challenge for our research group, since past sonification projects faced by the members [8-10] coped with smaller sets of information. In order to facilitate data-to-sound and data-to-process mapping tasks, we implemented our work in Pd. Although it made us to create a new data set adapted from the original one, it also allowed us to distribute the audiovisual processing in four Raspberry Pi (aka RPI) units.

The first section will examine the information contained in the original dataset, the characteristics of the one we created and the specifications of BuzuDados, the Pd abstraction which parse our dataset. The second section will report the data-to-sound mapping strategies implemented in the BUZU engine, by describing the behaviour of time, the drone sound synthesiser and the melody generation machine. The visual display and the multimedia hardware structure used in the first version of BUZU as well as the context in which it was exhibited will be discussed in the third section. The last section will outline some conclusions and future work.

2 Parsing the dataset

2.1 Analysing the original dataset

The original dataset and its documentation are available in the InterSCity website [1], [11]. The “Bus movement model”, as it is named, is a 146Mbytes file with 8 pairs of XML files intended to be incorporated in the InterSC-Simulator [12]. It represents trips performed by 2,183 bus lines in São Paulo. The information feeding the dataset

came from two sources: GTFS (*General Transit Feed Specification*) and AVL (*Automatic Vehicle Location*). The former reflects the service planning by providing data such as bus line code, route, bus stops location and pathway between stops. These open data can be consulted on the SPtrans public site [13]. The latter reflects the system's real behavior by providing data such as departing time, departing frequency and average speed. These data are gathered by GPS devices mounted on each vehicle, and were provided by Scipopolis [14], a startup devoted to process São Paulo transportation data from the Olho Vivo system [15].

Seven of the eight pairs of XML files correspond to a single day of a "typical" week (22 to 28 October, 2107), that is, a week in which the system operated without interruptions. The extra pair corresponds to an atypical day, such as the Easter holiday, when the system presents anomalous behavior.

The first file of each pair, called *buses.xml*, comprises GTFS data, while the second one is called *maps.xml* and comprises information from both sources. The two XML files interact each other, since in the *buses.xml* the bus stop locations are coded as a ten digits long number called node, while in the *maps.xml* every node corresponds to a couple of Lat and Long coordinates. It means that the software exploring the dataset should be able to shift between the two files in order to find data related to a single bus line on a single day.

2.2 Creating the NuSom dataset.

For BUZU we adopted Pd as the main software tool, since it can be run on the RPI. It would keep the project's costs lower and would allow us to bring the Sao Paulo transportation system's data to an open sound-making arena. With this in mind, our next step was trying to parse in Pd information from the InterSCity dataset.

We chose bus stop locations as the first data to be retrieved because they can be checked on online platforms such as Open Street Maps [16]. This empirical method were helpful to verify whether large amounts of parsed data were consistent with other maps.

Our first try was with the POF external library [17], which has an xml-parsing object. However, we failed retrieving the data since the XML files seemed to be too large and their arriving time was rather erratic. Another problem we faced was data loss. Pd (0.49-0) have a single precision floating point of 32 bits, which truncates large numbers such as those concerning with bus stop locations. In view of these difficulties we opted for creating our own dataset, to be used in Pd with the help of other programming tools. The limitations

with single precision of Pd have been discussed [18] in the Pd developer community. The solution for a double precision resolution has not yet been adopted by the issue of code compatibility between earlier versions and externals produced by the community. Our solution here was adopt other programming techniques to adapt the dataset to fit with Pd capabilities.

Inside Pd's parsing possibilities we started trying to use the Cyclone library *coll* object [19] to host data from the xml documents. When the tests started to fill the object buffer with large amounts of data we began to identify some data loss so we start to use the *text* object which suits better for the task and keeps the compatibility with Pd vanilla.

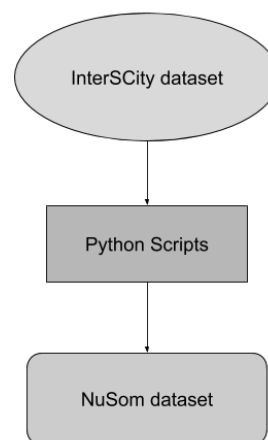


Figure 2: Dataset diagram

With Python 3 and the Pandas library we created three scripts that extracts some information from the originals *buses.xml* and *maps.xml* as well as from other intermediate files created by us, and distribute these data in a new set of files. Our goal was that the new dataset, named NuSom dataset could be readable by the *text* object.

The first python script, *cria_onibus_dia_pd-vanilla.py* generates 1 file per day. It parses the 8 original *buses.xml* and creates 8 new files. The former was created to adapt the new dataset to the *text* object operation together with *list split* object. The dataset display GTFS data such as departure interval, start time and bus stop locations. These last data were successfully retrieved appending *id* characters to the original longer integers. This bus stops identifier prevented from data loss and resolved the precision floating-point issue.

The second script, *cria_trajetos_dia_pd-vanilla.py* generates 1 file per day. It retrieves data from the *map.xml* files and creates 8 files. They are named *trajetos* and display the total distance in the course in meters and average speed at a set time once a trajectory identifier has been provided. Most of these identifiers correspond to bus lines

trajectories, which consists of the first and last bus stop separated by a dash.

The third script `cria_coordenadas_dia_pd-vanilla.py`, works in a different way. The script generates 8 files named `coordenadas` by retrieving information from some files created for us in an earlier phase of the project. These files are called `map_id_x_y-latlong`, and were obtained manually from the `map.xml`. We got the Lat and Long coordinates by converting the original nodes coded in UTM (Universal Transversal de Mercator) using an online converter [20] and a flexible text editor. Then each Lat and Long coordinate was scaled from 0 to 1000 in order to enable the creation of a meaningful file for sonification. By associating the correct bus stop identifier (the large number + the `id` character), the script `cria_coordenadas_dia.py` creates pair of scaled coordinates for each bus stop. It resolved the single precision floating-point issue. Furthermore, the script should be able to retrieve city region data from the postal code system, but this feature is under implementation.

2.3 BuzuDados specs

To navigate in our dataset we patched the `buzuDados` abstraction. It has two inlets and two outlets and works by demanding information with messages sent to the left inlet. Data is retrieved in the left outlet. `buzuDados` starts working when sending a number from 0 to 7 in the right inlet corresponding to the day of the week or the Eastern holiday. The right outlet will bang after data is retrieved. The left inlet and outlet works as follows.

When sending a number from 0 to 2.183 corresponding to the `lista_onibus` file, `buzuDados` retrieves the string `<bus id>`, which corresponds to the unique bus line identifier according to the SP trans system. When sending a message with `<bus id>` and then `<start_time>`, `buzuDados` returns a string symbol corresponding to the time in which the bus line should start operating; `<bus id>` and the string `<interval>` a list of 24 numbers, each one corresponding to the departure frequency (in seconds) for each of the 24 hours of the day; `<bus id>` and the string `<stops>` a list of strings corresponding to the identifiers of each bus line stop.

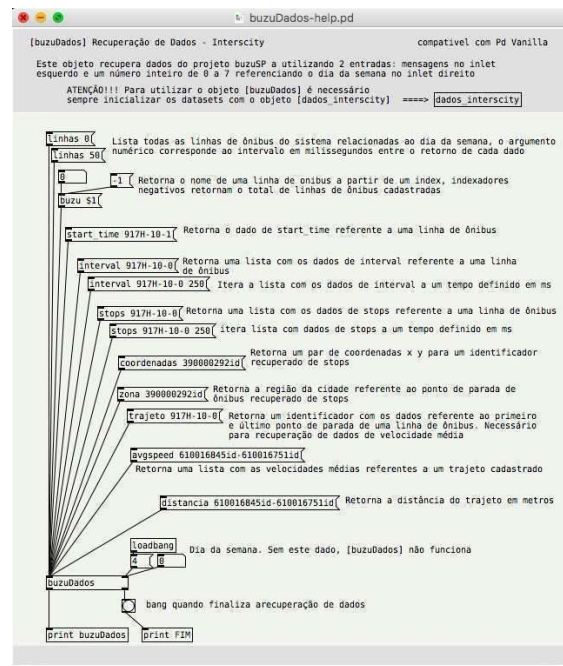


Figure 3: The `buzuDados` help.pd

When sending a message the string `<coordinates>` and a bus stop identifier, `buzuDados` retrieves a list of 2 numbers `<x>` and `<y>` corresponding to the scaled Lat and Long coordinates of this bus stop. When sending `<zona>` and then a bus stop identifier, it retrieves a number from 0 to 4 corresponding to São Paulo zones: Downtown, Western, Eastern, Northern or Southern.

Lastly, when sending a message with the string `<avgspeed>` followed by the first and last bus stop identifiers of a line separate by a dash, `buzuDados` retrieves a list of 24 numbers corresponding of the average speed (in m/s) at each of the 24 hours of this day. The NuSom dataset and the `buzuDados` abstraction are open and can be downloaded in our GitHub repository [21].

3. Data-to-sound mapping

Simultaneously with the process of constructing and refining the dataset retrieval and analysis, we conducted a sort of experiment in the design of the sonic content. The goal here was finding the sound poetics suitable for exploring the passage of time within the public bus transportation system, which could be reflected in a listening environment. Although BUZU did not implement any interaction device, it does provide a synesthesia experience between the projection of the bus lines highlighted on the map and the sound being generated. The sound mapping strategies were defined taking into account the possible struggles that people could experience during a bus trip. This would be experienced by contemplating

the data visualization and sonification present in the installation.

The BUZU audio engine is composed by two synthesisers working simultaneously: the drone and the melody maker machine. The former operates a an acoustic background, a lower-spectrum pad sound giving the sensation of continuity. It retrieves AVL data. The latter sonifies the route of up to four randomly chosen bus lines, by tracking the path followed at each bus stop. It produces identifiable melodies and retrieves mainly GTFS data. The drone and the melody maker machine receive data from `buzuDados` abstraction and are driven by the transport device.

3.1 Time

Our strategy to manage the passage of time in BUZU was using a Transport device, which is a time management interface widely used in DAW (Digital Audio Workstation) devices, audio recording and editing software. It is in the transport that can be found the play, pause and stop buttons. Once activated, a count is started by the Transport device with precision of 100 milliseconds. Through successive divisions, a count is generated where we have output such as day, hour, minute, second and millisecond.



Figure 4: BUZU transport device

It is possible to change the day and the speed. These features facilitate navigating the dataset over time, making possible to select the day and time when the dataset is queried.

3.2 The drone synthesiser

The drone synthesizer sonifies `<distance>` and `<avgspeed>` data, referring to the bus line pathway contained in the NuSom dataset. The strategy adopted was to use three overlapping textures to generate a predominantly contemplative sound structure.

The first texture is a sort of subwoofer, we used this layer in order to establish a synthetic and timeless acoustic space for the installation. We employ the `lfnoise~` generator and `else` library filters in order to get a subtle and deep texture. Since the audio system is quadraphonic, we chose the low frequency noise generator with fixed seed

in order to produce monophony between the two pairs of stereo audio systems. The existence of this layer, is independent of the dataset, since it fulfills a role of aesthetic establishment of the acoustic space that the work intends to install.

The second texture is similar to the Risset cascade of arpeggios [22]. By implementing additive synthesis, we generate a sonic spectrum that, through small alterations of pitch, is able to modulate its harmonics and promote sound beats.

In this texture we intend to produce states of restlessness and relaxation by manipulating the internal beat rate of the sound spectrum according to the average speed of the bus line at a given time of the day. This rate is obtained by calculating the weighted harmonic average [23] of the `<avgspeed>` in the complete route (from the first to the last bus stop) of a given bus line. The weighted harmonic mean velocity is calculated by making use of the average speed data of each path at a given time and the distance of each bus line path. The ratio between the average bus line speed in the time of day and the weighted harmonic mean of the line speed in the full path generates the number representing the average speed variation rate which, after being scaled, will control the density/velocity of the beats between harmonics in the second texture.

In this way, the slower the movement of the specific bus line, the greater the interference of sound beats in the generated tone. This mapping seeks to mimic in the sound plane a slow and possibly uncomfortable passage of time inside a bus in a busy traffic. The movement of the harmonics in this model generates very noticeable results, and potentially communicate the density of traffic in the sound domain.

Lastly, there is a noise texture whose morphology is dependent on the average speed of each bus line and can only be heard when invalid data is retrieved. In the InterSCity dataset, the `<avgspeed>` data with a value of -1 indicates that some data sampling failure occurred (failure to transmit, receive or even fail to trigger the GPS), in the NuSom dataset this value. Thus, by having an average speed in the path of -1, the noisy texture will be activated. The average speed of the bus line in the full path feeds the density parameter of the `dust` generator present in the `else` library [24]. It will generate a grainy random texture and can refer to an idea of analog noise, directly signaling a failure in the original dataset.

3.3 The Melody Maker Machine

The melody maker machine is a four-voice polyphonic FM synthesiser attached to a resonant filter using the `bob~` Pd object, and then attached to

a spectral delay [25]. The synth is driven by a dynamic ADSR, which receive pulses coming from the Transport device. It has four parameters (pitch, cutoff frequency, resonance, and delay feedback) and four outputs envisaging the quadraphonic arrange.

The synth retrieves information such as <coordinates>, corresponding to the GPS location data of the bus stops achieved by each line in <x> and <y> axes, and <interval>, corresponding to the number of departures per line at each hour of the day. The former was mapped to spatial data by producing an acoustical matrix of 1000 x 1000 values, the latter to the delay feedback parameter.

While our goal was creating an acoustical image of the city, our task was emphasize a sense of direction to each cardinal point rather than an exact location, by making evident direction changes. Thus, we assume a divergent mapping or many-to-one technique, where "...objects usually change their sound characteristics in several aspects at the same time when varying". [26]. In this regard, the <x> and <y> coordinates of each visited bus stop were mapped in two different ways.

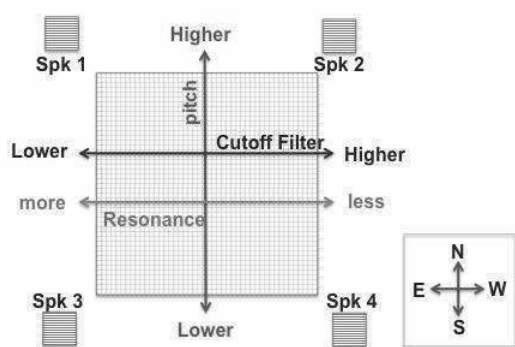


Figure 5: Mapping the <x> and <y> coordinates in the melody maker

On the one hand, the <x> coordinate was scaled and connected to the Cutoff frequency and, in reverse order, to the Resonance parameter. The <y> coordinate, connected to the pitch value, was scaled and redirected in order to avoid chromatic relations by selecting just notes of the Cm pentatonic scale. On the other, the coordinates where connected to the audio output levels, taking advantage of the quadraphonic system to recreate the cardinal directions in the installation space. The east-west axe was associated to the <x> coordinate, and the north-south axe to the <y> one.

4. The BUZU Auditory Display

4.1 Complementary visualization

In BUZU a screen is in the center of the quadraphonic space showing the spatial

displacement of the bus lines on the map representing the metropolitan region. The map is rendered with the Processing language connected via OSC to the server buzuDados (running in Pd).

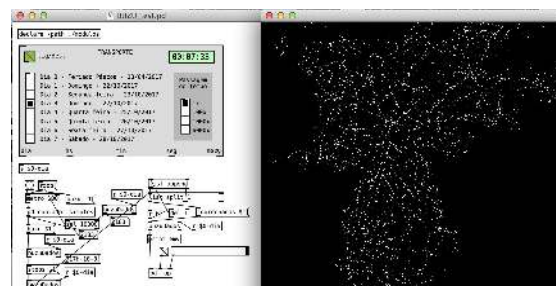


Figure 6: Processing visualization receiving data from the server

Around the central screen there are 4 LCD display connected via the wi-fi Wemos D1 mini microcontroller. The displays showed a text corresponding to the number ID of each bus line highlighted in the map. The main purpose with this visual clues was to provide a reference that triggered the recognition of the acoustic parameters, and then the emergence of a sounding image of the city.

4.2 Distribution of multimedia tasks

BUZU distributes audio synthesis, image generation, and data analysis tasks to four RPI 3 B+ units over a Local Area Network (LAN).

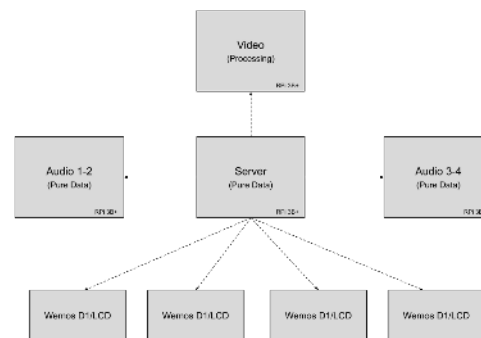


Figure 7: Buzu operational diagram

A RPi play the role of a server, in which the Transport is running, the NuSom dataset is parsed, and the data for sound and image synthesis is generated. The data is sent over the WiFi LAN, using the UDP-OSC implemented in the Pd *netsend* object. The data is sent through the router's broadcast address, to the other 3 RPi and the 4 WiFi Wemos D1 mini microprocessors. There are, in sum, seven clients.

The Audio 1-2 and Audio 3-4 RPi units are responsible for quadraphonics, with Audio 1-2 being responsible for the front stereo pair and

Audio 3-4 for the stereo rear pair. The Video RPi is responsible for the screen real-time rendering of the bus stop data used in the sonification and finally the Wemos D1 mini are responsible for receiving and displaying the ID of each retrieved bus line

5. Conclusions and future work

By taking advantage of sonification techniques BUZU offers a poetic experience in the perception of the passage of time by contemplating urban traffic. While visiting the installation, it is possible to experiment a kind of poetic enchantment by the contemplating the complexity of the system. At the same time, an image of the city start emerging when the visitor realize the relation between sound and the dynamic map. In this regard, the project goal was completed, since our intention was to generate an alternative view of Sao Paulo by retrieving data from the bus transportation system

However, although the installation worked very well in its public exhibition, and, the performance of the *buzuDados* abstraction retrieving data from *NuSom* dataset run well, we can envisage some adjustments. One of them is concerned with the feature capable of create interruptions in the calls made to the dataset. In addition we plan to implement internal messages in the control of the *buzuDados* dataflow. It will enable the expansion and maintenance of data parsing features in a more robust way, facilitating the abstraction implementation by third parties.

Although *buzuDados* completes specific operations for BUZU, it deals with processes relevant for other sound designers concerned with big data sonification. *buzuDados* makes feasible for others the implementation of the *InterSCity* dataset in Pd. It is expected to create alternative sonifications of the *NuSom* dataset using the *buzuDados* abstraction in collaboration with other members of the research group. Furthermore, a closer collaboration with *InterSCity* members is also planned, regarding the implementations of other datasets.

Acknowledgments

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9.

References

- [1] *InterSCity*. *InterSCity Website*. <http://interscity.org/>
- [2] Macêdo Batista, D; Goldman, A; Hirata, R Jr.; Kon, F; Costa, F and Endler, M. (2016) *InterSCity: Addressing Future Internet Research Challenges for Smart Cities*. 7th IEEE International Conference on Network of the Future.
- [3] *Sons de Silício*. (2019). *Art Exhibition Website*. <https://sonsdesilicio.wordpress.com/>
- [4] Nickerson, L.V., Stockman, T., & Thiebaut, J. (2007). *Sonifying the London Underground Real-Time-Disruption Map*. *Proceedings of the 13th International Conference on Auditory Display*, Montréal, Canada, June 26 - 29, 2007
- [5] Adhitya, S & Kuuskankare, M. (2011). *The Sonified Urban Masterplan (SUM) Tool: Sonification for Urban Planning and Design*. *Proceedings of the 17th International Conference on Auditory Display (ICAD-2011) June 20-24, 2011*
- [6] Brittell, M. (2018). *Seeking a reference frame for cartographic sonification*. *Proceedings of the 24th International Conference on Auditory Display June 10 -15, 2018*
- [7] Gover, M. (2018) <https://www.matangover.com/metrosynth/>
- [8] Arango,J,J. (2018) *AirQ Sonification as a context for mutual contribution between Science and Music*. *Música Hodie*. V.18.N.1. pp: 92-102. DOI:10.5216/mh.v18i1.53573
- [9] Arango,J,J. (2018) *Esmog Data: Interpreting Air Quality Through Media Art and Design*. *Leonardo Electronic Journal ISSN: 1559-0429 ed: v.51 fasc.N/A p.183 - 183 DOI: https://doi.org/10.1162/LEON_a_01515*
- [10] Iazzetta, F; Piccinini, A. (2018) *Sonificação como art-science: três lógicas de interdisciplinaridade, três lógicas de sonificação*. *Anais do XXVIII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*. Manaus.
- [11] Wen, M. et al.(2018) *Criação de Modelo para Simulação de Movimentação de Ônibus a Partir de Dados Reais*. *I Workshop Brasileiro de Cidades Inteligentes*.
- [12] Santana, E. F. Z.; Lago, N; Kon,F and Milojevic, D. S. (2017) “*Interscsimulator: Large-scale traffic simulation in smart cities using*

erlang,” in The 18th Workshop on Multi- agent-based Simulation

[13] SP Trans. Website

<http://www.sptrans.com.br/desenvolvedores/GTFS.aspx>

[14] Scipopulis Website.

<https://www.scipopulis.com/>

[15] Olhovivo Website.

<http://olhovivo.sptrans.com.br/>

[16] Openstreet Maps Website.

<https://www.openstreetmap.org>

[17] POF - Pd + Openframeworks.

<https://github.com/Antlr/ofxPof>

[18] Katjaa, V. Double precision in Pd. 2011.

<http://www.katjaas.nl/doubleprecision/doubleprecision.html>

[19] Cyclone library. Repository.

<https://github.com/porres/pd-cyclone>

[20] Leware Website. UTM to LatLong converter.

<https://leware.net/geo/utmgoogle.htm>

[21] Viveros, E. Arango, J. J. & Iazzetta, F. NuSom dataset.

https://github.com/LABNuSom/Interscity_pd_data

[22] Risset, J. C. (2003). Computer Music: Why?

https://liberalarts.utexas.edu/france-ut/files/pdf/resources/risset_2.pdf. Retrived 2019-05-26. Online pdf document.

[23] Wikipedia. Harmonic mean. Acessado em: 28/05/2019.

https://en.wikipedia.org/wiki/Harmonic_mean#Weighted_harmonic_mean

[24] Porres, Alexandre Torres. Else Library for Pure Data. 16th Brazilian Symposium on Computer Music. 2017.

<http://compmus.ime.usp.br/sbcm/2017/papers/sbcm-2017-6.pdf>

[25] Barnecht, F. (2012) Guitar Spectral Delay.

<https://guitarextended.wordpress.com/2012/02/07/spectral-delay-effect-for-guitar-with-pure-data/>

[26] Grond, F&, Berger, J. (2011). Parameter Mapping Sonification. In: Herman, T., Hunt, A & Neuhoff, J. G. The Sonification Handbook.