grainBirds: Evolutionary granular synthesizers distributed in Fog Computing

Jônatas Manzolli[1 2 *], **Edelson Henrique Constantino**[2]

[1]Arts Institute – University of Campinas
Rua Elis Regina, 50 – Cidade Universitária ”Zeferino Vaz”, Barão Geraldo, Campinas, SP

[2]Interdisciplinary Nucleus of Sound Communication - University of Campinas
Rua da Reitoria, 165 – Cidade Universitária ”Zeferino Vaz”, Barão Geraldo, Campinas, SP

jonatas@nics.unicamp.br, edelson@nics.unicamp.br

Abstract. *The article presents the grainBIRD concept inspired by a distributed fog of grain sounds metaphor. Shortly, granular synthesizers are distributed in a mobile device's network. Applying Interactive Genetic Algorithm (IGA) and exchanging Open Sound Control (OSC) messages in a Virtual Private Network, grainBIRDs generate sound grains in standalone and network configurations. The grainBIRD's concept dialogues with the Internet of Things (IoT) paradigm since it was inspired by Cloud and Fog computing architectures. In this sense, this article presents the grainBIRD and grainBIRD Orchestra concepts, followed by a review on Computer Music and Network technologies related to the project. The article also discusses the computer implementation of the grainBIRD application using the Pure Data programming environment and concludes with performance tests of the network communication feasibility and the system sound generation capacities.*

1. Introduction

Given the challenge of creating and interacting with other artists remotely caused by the COVID-19 pandemic, grainBIRD was created. It is possible to say that the grainBIRDs establish a network to share sound control parameters from one place to another: the flighting of messenger birds. The grainBIRD's concept was born in a series of musical letters called cartas@todocanto. Letters were sent to musicians and dancers [1] in different geographic locations, in Brazil and abroad. Shortly, grainBIRDs carry parametric sound letters whose exchange among devices generates a fog of sound grains. The network interaction was projected as a suitable process for distant co-creation among musicians and dancers. Particularly, the use of the internet as new environment offering to composers and creators a medium for collective creation and production of open and continuously evolving works, as discussed by [2].

The implementation of grainBIRD brought together different techniques from the Computer Music literature such as the contemporary notion of laptop orchestras [3, 4, 5] and also the Ubiquitous Music perspective [6]; the granular synthesis [7], [8] used to implement grainBIRD's synthesis engine. Further, the mobile granular synthesizers operate with Interactive Genetic Algorithms (IGA) [9, 10, 11, 12]. grainBIRDs communicate in two ways: a) with each other in a network of mobiles devices, and b) with desktop host computers. Thus, grainBIRD network implementation dialogues with the Internet of Things (IoT) paradigm [13], Cloud Computing [14][15], and Fog Computing [16].

Communication among grainBIRDs is carried out by exchanging Open Sound Control (OSC) messages [17] between local devices, such as the fog infrastructure concept, and over the Internet through the Virtual Private Network (VPN) [18] interface. Considering that all devices will communicate locally or remotely, OSC messages are exchanged through a multicast group. Figure 1 introduces the network communication architecture in line with the following concepts:

> **grainBIRD:** *application for a mobile device that runs a granular synthesizer controlled by IGAs. The synthesis is controlled locally or by OSC[1] messages exchanged in a grainNET.*

> **grainBIRD Orchestra or grainNET:** *Orchestra formed by communicating the different grainBIRDs with each or with a host within a VPN[2] network.*

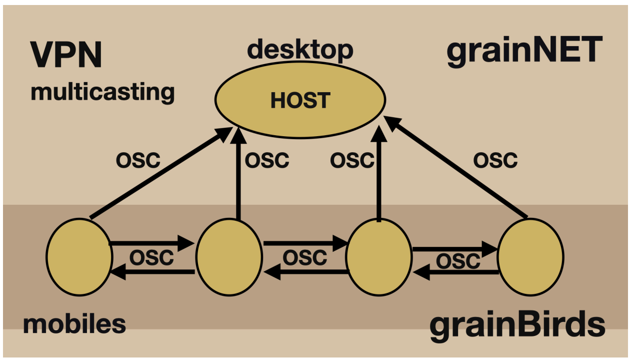

Figure 1: Diagram illustrating the concept of grainBIRDs in a network.

*Supported by CAPES.

[1]http://opensoundcontrol.org

[2]It is a VPN that extends a private network by a public network allowing users to send and receive data over public or shared networks, as if their devices were connected to the private network directly.

2. Dialogue with Computer Music Technologies

Networked musical performance might date back to a concert performed on July 3, 1978 at Blind Lemon, a small music gallery in Berkeley, California [19]. It could be the first collective performance of a composition distributed in a microcomputer network. Later, in 2005, the Princeton Laptop Orchestra, an ensemble of fifteen laptop-based meta-instruments, began its first season [5]. The possibility of using distributed systems in many embedded devices available for musical composition and performance dialoguing with IoT technologies, is discussed in [13]. grainBIRD dialogues also with the ubiquitous music perspective [6], in a previous work we described the interaction and collaboration in computer music using networks [14].

2.1. From Laptop Orchestra to grainBIRD Orchestra

A laptop orchestra or laptop ensemble is a chamber music ensemble consisting primarily of the collective use of laptops as musical instruments. With artistic and educational purposes, these orchestras focus on creating innovative technologies and artworks that use improvisation and frequent collaborations with other ensembles and soloists, dancers, video artists, and actors [4].

In the study presented here, we implemented a derivation of the laptop orchestra concept. Besides, the grainBIRDs provide means for local performances, distributed performance with mobile devices, or performances merging local ensembles with virtual ones.

2.2. From Granular Synthesis to Grain Fog

Granular synthesis (GS) was born with Dennis Gabor's concept of a "sound quantum" in 1947 [20]. Nowadays, it is defined as a digital synthesis operating in the time scale of microsounds. It is based on sequencing samples that are partitioned into small segments called "grains." Specifically, these grains have durations less than tenths of a second and great than tens of milliseconds [7].

In the project discussed here, we draw an analogy between fog computing and the fog of granular sounds generated by mobile devices. A grainBIRD contains sound samples modulated by envelope curves and parameters such as the start reading point, duration, pitch, and overlapping rate of each grain. These five parameters are controlled by IGA, as presented next.

2.3. Interactive Genetic Algorithm in a Network

Digital sound synthesis is a natural domain for applying evolutionary computation [11]. The concepts of the genome and genetic operations can be consistently mapped into synthesis parameters [11]. In previous work, we have already introduced the concept of evolutionary synthesis [10]. Another argument is that GS can give rise to unexpected and complex changes in the sound output, and with Genetic Algorithms (GA), it could be possible to deal with that within a semi-automatic approach, using human intervention. The IGA is also a general term for GA that uses human evaluation as a fitness function [9]. In other words, the synthesis process follows the user's particular preference. We have also discussed IGA with the concept of ArTbitration [15], which denotes the users' aesthetic judgment using evolutionary computing. A way to interact with sound processes and increase the complexity of the resulting sound without neglecting aesthetic aspects.

The application of IGA was relevant for our project because grainBIRDs are controlled explicitly from the graphical interface and implicitly from motion sensors. The coupling between GS and IGA provides a fertile field for exploration. Digital sounds would be generated with vitality and musical interest in a wide range of temporal scales and timbres.

3. Dialogue with Network Technologies

The IoT concept is guided by creating a dynamic network infrastructure to connect anything, which can be physical or virtual objects, through different means of communication in which each device receives a unique IP address. Therefore, the devices have characteristics such as identity, physical attributes, positioning sensors, monitoring, and other sensors that collect some type of information. Such information is further communicated and controlled through the Internet [21]. In the grainBIRD's network concept, the access through a VPN and the OSC protocol enables the morphology control of a distributed fog of sound grains.

3.1. Internet of Things

A grainBIRD comprises what we define in IoT as devices that perform one or more operations within a specific or dedicated scope. The grainBIRDs communicate and exchange messages with the other devices present in the communication infrastructure through IP and OSC protocols.

In order to allow efficient communication among grainBIRDs, the established and response time (latency) among the devices must be in the desired minimum value (see tests performed below). When executed in local infrastructure, time could not be a limiting factor. However, as in grainBIRDs, time latency becomes a significant challenge when it operates in a distributed infrastructure.

Given the above, we experimented with cloud and fog computing paradigms to create a high-performance digital infrastructure to enable connection and communication between grainBIRD devices. We looked for local and remote latency that could make it possible to exchange granular synthesis parameters, trigger, and other specific controls.

3.2. Cloud Computing

Cloud computing enables the development of solutions and technologies without significant hardware or skilled labour to support and maintain services infrastructure [22]. Additionally, Cloud Computing offers agility in the supply of hardware, better use of computing resources, and the possibility of redefining resources at any time, enabling the development of projects requiring high performance and dynamic resource allocation.

The National Institute of Standards and Technology (NIST) defines Cloud Computing as "a model that enables ubiquitous, convenient and on-demand access to a set of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be quickly provisioned and released with minimal management effort or service provider interaction" [23]. In this context, the computational resources are made available and accessed through the Internet from the existing network infrastructure on site [24].

In order to meet the communication requirements between grainBIRD devices, local and remote, we explored the cloud and fog computing paradigms, which has, among its characteristics, the expansion of the local network coverage infrastructure to ensure stability and high performance, such as low latency, as required for the performers' interaction and the sound synthesis control shared among grainBIRDs. The fog computing concept is presented next.

3.3. Fog Computing

Applications that require low latency, high availability, or mobility can be hampered in locations or regions where coverage and bandwidth are limited or even non-existent. The fog computing paradigm is linked to the implementation of cloud computing resources at the edge of network infrastructures, such as data storage or processing. According to Yi et al. [25], fog computing provides dynamic resources and services to users at the edge of the network, while cloud computing provides resources distributed across the core network.

For Nandyala and Kim [26], fog computing can "act as a bridge between smart devices and large-scale cloud computing and storage services." In the present project, we define fog computing as a decentralized architecture in which grainBIRD devices provide information and communicate through the digital infrastructure built between the network edge and the Cloud.

As shown in Figure 2, grainBIRD devices communicate locally and remotely via the grainNET network infrastructure (i.e., the grainBIRD Orchestra). This network is based on IoT – devices connected locally that compose a haze for communicating with each other via cloud computing. In this setting, they communicate and interact with other remote devices when connected to the grainNET network.

In short, grainNET is a digital infrastructure that permeates the connection between various grainBIRDs' fog – which is integrated into the device cloud. We understand the sound generated by the grainNET in two ways: 1) a texture generated locally by the grainBIRDs and 2) the texture generated in a host computer by the sum of the OSC messages send by grainBIRD devices in a VPN interface and multicasting manner.

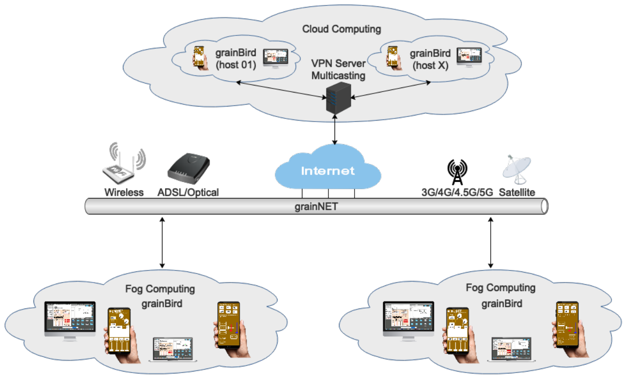

Figure 2: Digital infrastructure for communication between the Cloud and Fog among grainBIRDs.

4. grainBIRD Implementation

The following paragraphs discuss the features and architecture of the grainBIRD implementation with Pure Data Vanilla[3] programming environment [16] and freeware MobMuPlat, created by Daniel Iglesias[4]. The grainBIRD's granular synthesizer was derived from Farnell [27].

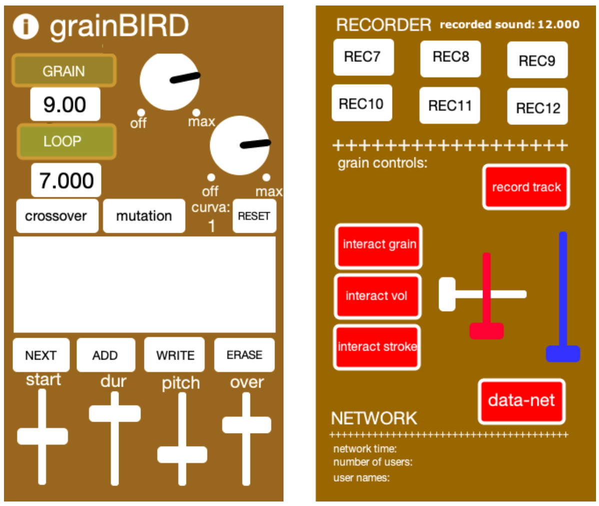

Figure 3: GrainBIRD GUI: granular synthesizer controllers (left) and toggles to control local recording and network communication (right).

4.1. grainBIRD application

The grainBIRD App allows the generation and transformation of sound grains. It is explicitly controlled with the GUI (see Figure 3) and implicitly by motion sensors. Using Pure Data programming language, we developed the grainBIRD App organized in four main groups of subpatches: 1) to play and generate sounds, 2) to execute IGA operations, 3) to store synthesis pre-sets, and 4) to control network communication. Figure 4 presents the grainBIRD App main patch.

[3] https://puredata.info/downloads/pure-data
[4] https://danieliglesia.com/mobmuplat

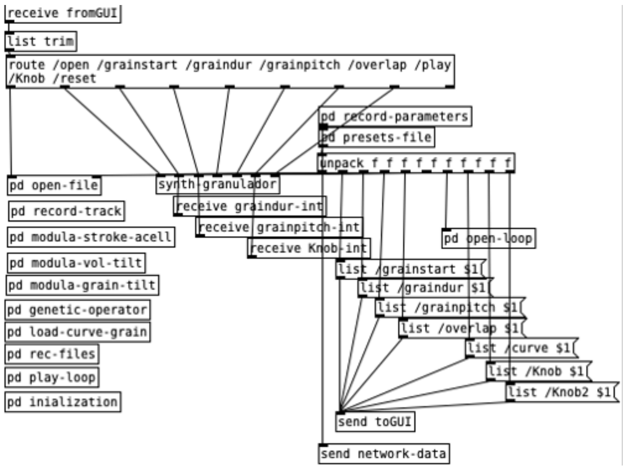

Figure 4: Main Patch of grainBIRD App.

4.2. Interactive Evolutionary Granular Synthesizer

The IGA implementation comprises a population, a genome, and two genetic operators: Crossover and Mutation (see [12] for more explanation).

PARAMETER	RANGE	UNIT
gen. operation	[-1,0,1]	
grain start	[0..1]	%
grain duration	[50..2000]	milliseconds
grain pitch	[0.1..2]	freq. ratio
grain overlap	[1..2]	
# Sound grain	[1..12]	
# Sound loop	[1..12]	
# Envelope	[1..12]	
grain volume	[0..100]	dB
loop volume	[0..100]	dB

Table 1: Parameters of the synthesis genome.

As described in Table 1, a 10-parameters list controls the granular synthesizer, the number indexes of sound grain, sound loop, envelope, and the volume of sound grains and sound loops. The IGA control of grainBIRDs follows this idea: mutation and crossover operations are controlled by local and remote players acting on the GUI or producing movement with their mobile phones. Thus, motion sensors, the accelerometer and tilt sensors, produce changes in the granular flow.

Another GS control is related to the shape of grain envelopes defined originally here as a 2048-points Hamming window, see implementation [27]. Moreover, the shape of these envelopes is also transformed by IGA in the following ways: 1) drawing the envelope on the GUI, 2) clicking on the crossover or mutation buttons to draw curves automatically, 3) mixing both.

Secondly, it is also possible to transform the envelopes by moving the mobile phone in two ways. The first one is associated with the Tilt sensor. When the user roll or yaw the mobile phone, the synthesis slider controls (see Figure 3) move together. Secondly, the crossover operation is triggered when the acceleration module related to the user's movement exceeds a threshold. Figure 5 displays two columns of three envelopes describing an automatic evolution starting upon the Hamming window and subsequently applying a crossover operation.

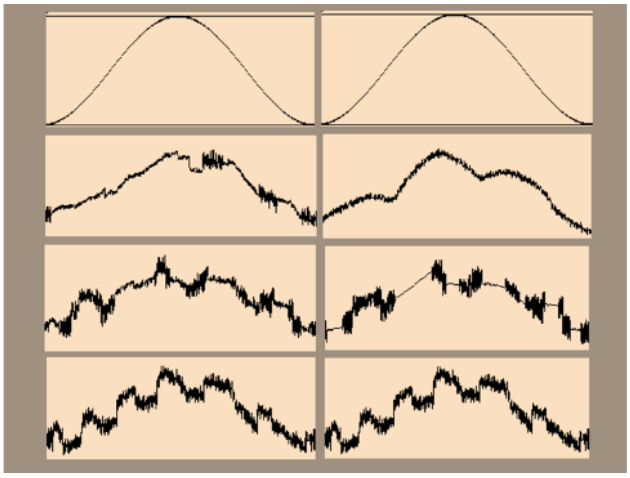

Figure 5: Automatic Evolution of grainBIRD envelopes from top to down: a hamming window is followed by 03 generations of envelopes.

The two types of evolutionary granular synthesis controls (which are composed of ten parameters and the envelope shapes) are stored in two files. Following the Evolutionary Computing paradigm, they are considered here as the grainBIRD genomes [11]. The first group, consisting of 12 files, contains the numerical values of the envelopes. The second file stores the 10-parameter list in each line as the user adds it to the file. In Figure 3, it is possible to see four buttons: Next, Add, Write, and Erase to control the file storage.

Finally, twelve sound samples are used to generate the grains, and the play sound loops are stored in WAV format of 16 bits and 44.100 Hz – the original grainBIRD implementation contains samples of Brazilian bird songs. It can be renewed in two ways: 1) storage a new group of twelve samples in the /sounds folder, and 2) six sounds can be recorded by the grainBIRD's app itself during a performance.

4.3. Network Communication

The flow of OSC messages [17] generated by the grainBIRDs enables interaction between artists in different remote locations. Each device produces its particular sound fog; they coexist in time and could be heard simultaneously or not. The system does not transmit audio in real-time, but only OSC messages among the network. Nevertheless, a performance with grainBIRD can be transmitted by a network audio distribution software enabling the coexistence of sounds in a synchronized texture.

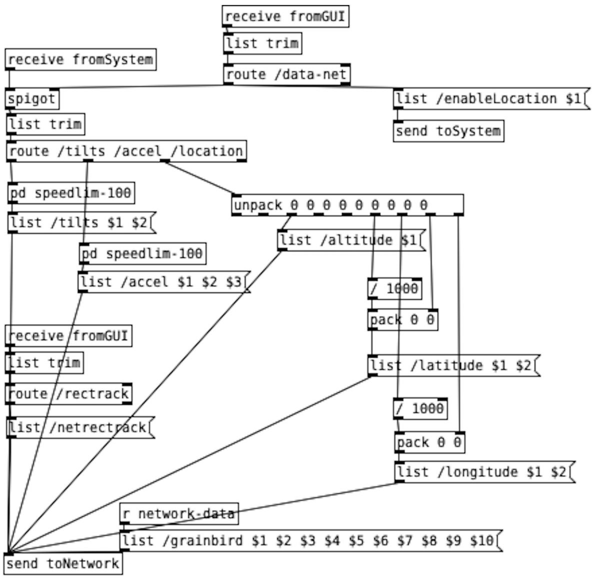

Figure 6: Sub-patch describes the grainBrain OSC communication implementation.

As described previously, the grainBIRDs were implemented as local standalone synthesizers whose parameters are modified by an IGA locally and by OSC messages remotely. The 10-parameter control list is transformed into OSC messages, and they are propagated in a VPN multicasting manner [18]. In addition, each grainBIRD transmits the variation of the motion sensor (Tilt) and the accelerometer, altitude, and geo-location. This data is used on the host grainNET machine to control the granular synthesis automatically. We will also implement a graphic plot of the geo-distribution of the grainBIRD Orchestra when the performers are performing from different remote locations. Figure 7 shows the GUI of the grainNET host computer, which is a mirror of the grainBIRD App.

5. Performance Tests

We carried out some tests of grainBIRD's network latency using a VPN and sound capacities – to measure its feasibility in real-time performances. In this setting, standalone solo and distributed performances were evaluated.

5.1. Network Performance

The VPN [18] used for testing can be described as a logical network created on top of the existing private or public physical infrastructure to provide a logical digital infrastructure for equivalent communication. This logical communication infrastructure was established from grainBIRD device(s) to the central VPN server, which is installed, in our experiments, at the Interdisciplinary Nucleus of Sound Communication (NICS) at the University of Campinas (UNICAMP). The VPN system is part of the set of services and applications allocated in cloud computing, as shown in Figure 8. In our implementation, the VPN uses public-key cryptography (asymmetric) with combined 2048-bit RSA keys with digital certificates signed by the server, making it highly secure and reliable. In this format, keys are initially generated on the server and associated with grainBIRDs devices, guaranteeing integrity, authenticity, and security in connection and communication.

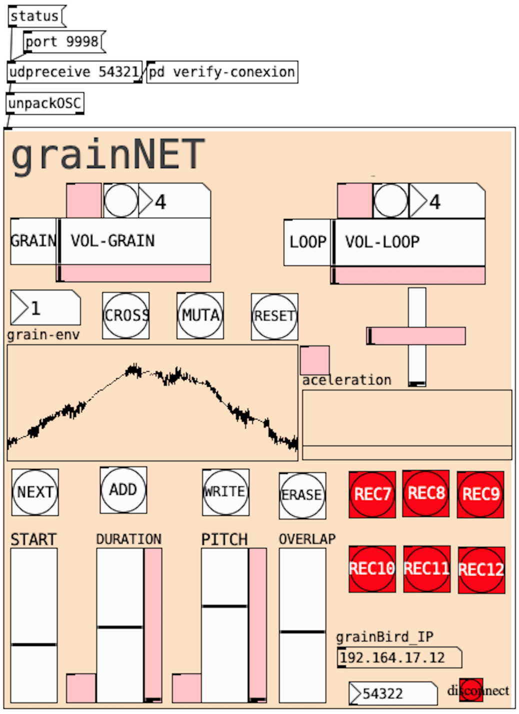

Figure 7: GUI of the grainNET Host computer.

As can be seen in Figure 8, after connecting to the VPN network and receiving the unique IP address, a grainBIRD starts to receive the messages sent through multicasting automatically. In addition, it was possible to view and communicate with the other grainBIRDs connected to the grainNET network. Therefore, through that bilateral communication, all grainBIRDs can send and receive messages through VPN, which was also defined in this project as a grainNET network (see Figure 2).

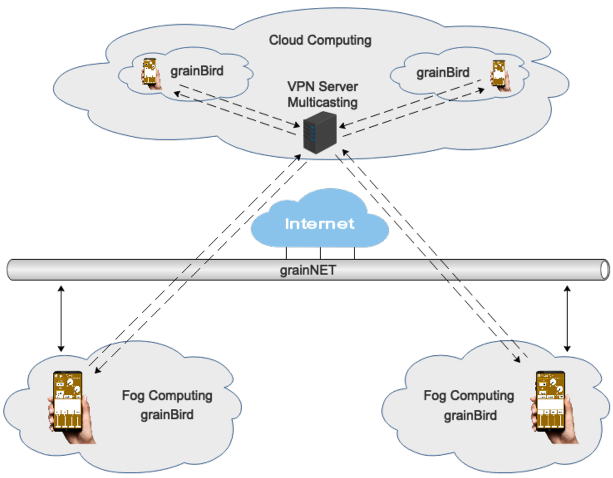

Figure 8: Multicasting communication over the VPN network used for testing the grainBIRDs.

Two experiments were carried out to measure and evaluate the communication among grainBIRDs: a) sending 100 packets (echo request/reply) between two local grainBIRD devices, and b) sending 100 packets (echo request/reply) between two grainBIRDs devices over the VPN network. As defined in this approach as fog, both devices were connected to the same network via a wireless router for the local test. For the remote test, one of the devices was connected to the VPN through the 4.5G mobile network provided by the operator Claro BR, and the second device through the home wireless network.

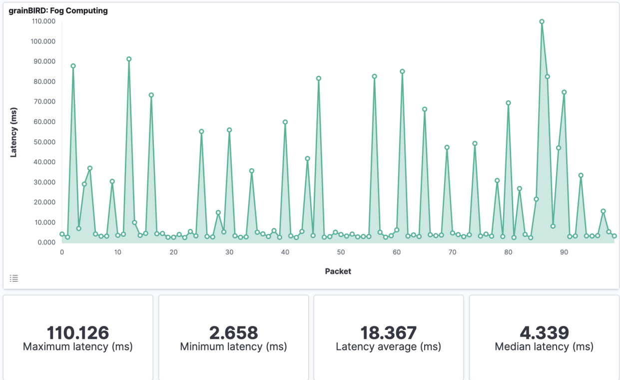

Figure 9: Graphic of the max, average, minimum, and median latency obtained in a local fog network.

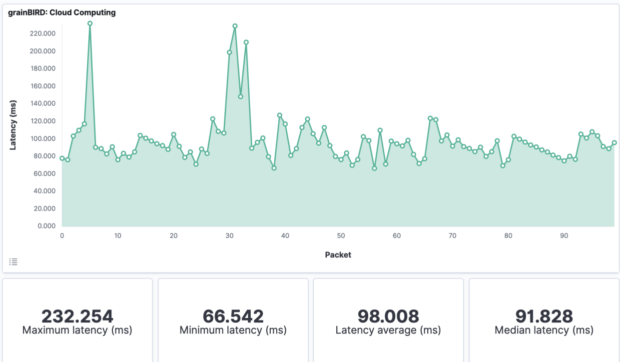

Figure 10: Latency max, average, minimum, and median latency obtained via the cloud VPN network.

Figure 9 shows the plot of latency obtained in the communication between two local grainBIRDs, that is, in the fog layer. Figure 10 shows the plot of latency obtained between two remote grainBIRDs, that is, communicating over the VPN network.

5.2. Sound generation performance

Firstly, we tested a solo performance just with one grainBIRD in a standalone situation. Figure 11 shows the audio track produced by the evolution of five envelopes while it was applied only to crossover operations.

Figure 11 also displays a sequence of five grainBIRD envelopes (in the top) align with the spectrogram (in the bottom). The vertical lines indicate changes on grain envelopes between approximately 20 seconds. Thus, it is possible to see the spectral content of the sound generated by that grainBIRD solo. It is important to note that the envelopes at the top of Figure 11 are applied to many grains of the granular synthesis (see Figure 5).

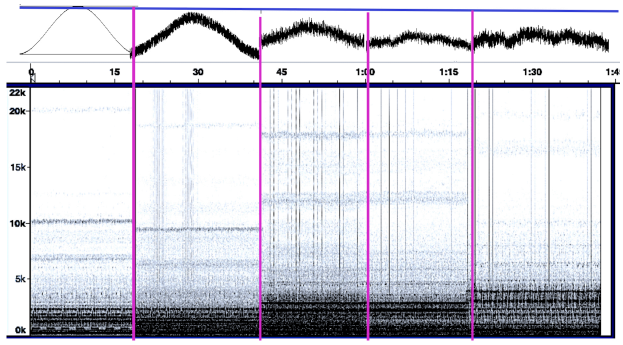

Figure 11: Recorded audio track during the grainBIRD solo test.

Secondly, we carried out a distributed performance test with two grainBIRDs and a host grainNET computer from a distance in a VPN environment. The grainBIRD devices and the host computer were synchronized with a start-play OSC message to record local soundtracks simultaneously. After that, OSC messages were sent over the network from a guiding grainBIRD so that the others responded together. It was sent from the grainBIRD1 05 crossover messages.

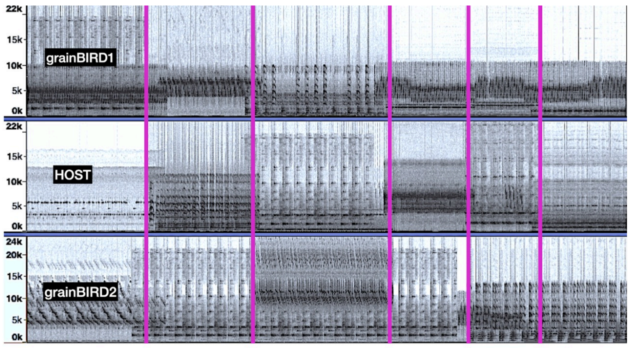

Figure 12: Spectrogram of the grainBIRD distributed performance.

In Figure 12, the local recordings of each mobile device are displayed together in three superimposed spectrograms. The top plot is from a grainBIRD1 run in an iPhone, the middle is from the Host grainNET run in a MacBook desktop, and the bottom is from a grainBIRD2 run in an iPad. The vertical lines indicate the grain envelopes synchronized changes over the network.

6. Discussions

From the graphics presented in Figures 9 and 10, it is possible to evaluate the latency range produced in a test exchanging 100 messages between two grainBIRD devices. Therefore, regarding the grainBIRD connection test, the latency obtained in the communication between the devices was an average of 18.367 ms, which is good network performance, and it meets the requirements of the grainBIRD devices. On the other hand, the latency obtained in the communication between remote devices during the tests through Cloud was on average 98.008 ms. This result was considered promising and allowed us to control and exchange messages between existing grainBIRDs in a remote fog network.

Regarding the solo performance test, it is possible to verify that the automatic evolution of grainBIRD envelopes can generate and transform the content of sound grains. The granular synthesizer of each grainBIRD already produces its sound fog, which is then associated with other grainBIRDs, thus forming a complex texture, as shown in Figure 11.

From the sound distributed performance test, the synchrony and complexity of the resulting soundtracks can be visualized (see Figure 12). The guiding device was used as a kind of orchestra director for grainBIRDs performed synchronized sound transformations. The network latency was not significant for generating sound grains as it happened quite simultaneously. It is crucial to notice that even there were transmission delays on sending the 10-parameters messages (see Table 1). This list controls granular synthesis macro while micro-controls are performed by each engine locally as defined above (see section 3.1 – grainBIRDs perform operations within a specific or dedicated scope).

Finally, to demonstrate the interaction with mobile devices motion sensors, we develop an improvisation test with three grainBIRDs and a host grainNET computer (see Figure 13). We use four different devices (iPhone, iPad, Moto G3) and a MacBook to test also compatibility between the App, different motion sensors, and OS of these different devices. The test consisted of two users in different remote locations over the Cloud, performing a movement with a device followed by the user in the other location and so on. The result presented in Figure 13 shows that the changes were always propagated through the network evenly and with a slight delay (i.e., see the sync demarcation of the vertical lines).

From the discussion here, it is possible to understand grainBIRD allows multiple performance configurations that can be hybridized using local and remote devices over a cloud or fog infra-structure and can also be part of several distributed performance configurations.

7. Conclusion

Given the social isolation caused by COVID-19 and motivated by the challenge of creating and interacting with other artists remotely, we created the concept of grainBIRD that was proof here. We imagine a haze of mobile devices generating granular sounds via OSC messages within a network. We implement a distributed granular synthesis process using a VPN network and IGA to interact between performers in remote locations. Thus, grainBIRD devices communicate and interact with others using remote devices connected to the grainNET network. Network communication was discussed here in light of the IoT, Cloud computing, and Fog computing paradigms. We

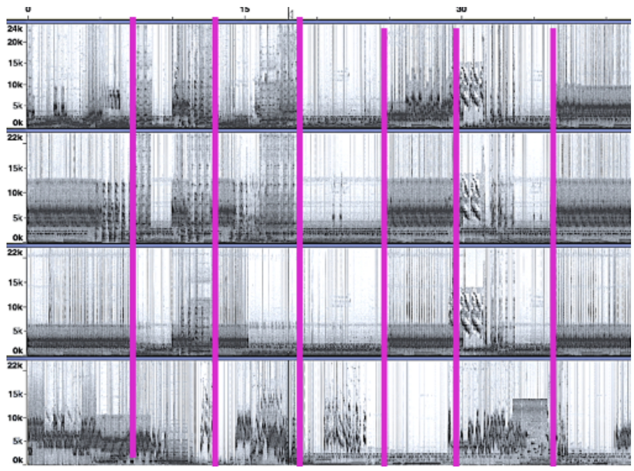

Figure 13: Spectrogram of the grainBIRD distributed improvisation with movement sensors.

showed that the access through a VPN and the OSC protocol enabled the control of the morphology of the sound fog generated by grainBIRDs.

The concept of grainBIRDs is derived from the concept of the laptop orchestra. However, we design the grainBIRDs' app with the notion of cloud and fog computing architecture, enabling both local interactions and performances over a network of mobile devices. In this sense the research presented here dialogues with the possibility of co-creation over the Internet [2] and also with the paradigm of Ubiquitous Music [6]. Therefore, we intend to use the geo-location of grainBIRD's devices to draw a distributed Orchestra map and control other sound parameters. We intend to develop also a communication application between the grainBIRDs and other digital musical instruments, thus providing what we could name as a network-distributed mixed music performance.

Finally, grainBIRD will be fundamental for developing the artistic research involving musicians and dancers, which motivated the development reported here.

Acknowledgment

We thank the CAPES-Print Unicamp project "Poéticas do som e do corpo no diálogo interdisciplinar entre arte e tecnologia: estudo de pesquisa de criação em Humanidades Digitais", 88881.311276/2018-01 and the Universal CNPq project, 429620/2018-7. Manzolli is also supported by the CNPq Pq grant 304431/20184.

References

[1] Jônatas Manzolli, Mariana Baruco Machado Andraus, and Milena Pereira dos Santos. Entre voos e quedas: (sobre)voos de pássaros (des)cuidados. *XV Simpósio Internacional de Cognição e Artes Musicais (SIMCAM)*, 2021.

[2] Sergi Jordà. Faust music on line: An approach to real-time collective composition on the internet. *Leonardo Music Journal*, 9:5–12, 12 1999.

[3] Jason Freeman and Akito Van Troyer. Collaborative Textual Improvisation in a Laptop Ensemble. *Computer Music Journal*, 35(2):8–21, 06 2011.

[4] Why a laptop orchestra? *Organised Sound*, 12(2):171–179, 2007.

[5] Ge Wang, Dan Trueman, Scott Smallwood, and Perry R. Cook. The Laptop Orchestra as Classroom. *Computer Music Journal*, 32(1):26–37, 03 2008.

[6] Damián Keller, Victor Lazzarini, and Marcelo Pimenta. *Ubiquitous Music*. 01 2014.

[7] Curtis Roads. *Microsound*, volume 148. The MIT Press, 2001.

[8] Barry Truax. Real-time granular synthesis with a digital signal processor. *Computer Music Journal*, 12(2):14–26, 1988.

[9] A. Kosorukoff. Human based genetic algorithm. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, volume 5, pages 3464–3469 vol.5, 2001.

[10] Jônatas Manzolli, Adolfo Maia, Jose Fornari, and Furio Damiani. The evolutionary sound synthesis method. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, page 585–587, Ottawa, Canada, 2001. Association for Computing Machinery.

[11] James McDermott, Niall J. L. Griffith, and Michael O'Neill. *Evolutionary Computation Applied to Sound Synthesis*, pages 81–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[12] Artemis Moroni, Jônatas Manzolli, Fernando Von Zuben, and Ricardo Gudwin. Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. *Leonardo Music Journal*, 10:49–54, 12 2000.

[13] Angelo Fraietta, Oliver Bown, Sam Ferguson, Sam Gillespie, and Liam Bray. Rapid Composition for Networked Devices: HappyBrackets. *Computer Music Journal*, 43(2-3):89–108, 06 2020.

[14] Flávio Luiz Schiavoni, Pedro H. de Faria, and Jônatas Manzolli. Interaction and collaboration in computer music using computer networks: An ubimus perspective. *Journal of New Music Research*, 48(4):316–330, 2019.

[15] Artemis Moroni, Fernando Von Zuben, and Jônatas Manzolli. ArTbitration: Human-Machine Interaction in Artistic Domains. *Leonardo*, 35(2):185–188, 04 2002.

[16] Miller Puckette. Pure data: another integrated computer music environment. In *in Proceedings, International Computer Music Conference*, pages 37–41, 1997.

[17] Dave Phillips. An Introduction To OSC, 2008.

[18] Ayhan Erdoğan and Dz Yzb. Virtual private networks (vpns): A survey. 01 2008.

[19] John Bischoff, Rich Gold, and Jim Horton. Music for an interactive network of microcomputers. *Computer Music Journal*, 2(3):24–29, 1978.

[20] D. Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159:591–594, 1947.

[21] M Divyashree and H G Rangaraju. Internet of things (iot): A survey. In *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, pages 1–6, 2018.

[22] Michael Armbrust, Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53:50–58, 04 2010.

[23] Tim Mell and Peter Grance. NIST Definition of Cloud Computing. *National Institute of Standards and Technology*, 2011.

[24] Flávio Sousa, Leonardo Moreira, and Javam Machado. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. 2009.

[25] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata '15, page 37–42, Hangzhou, China, 2015. Association for Computing Machinery.

[26] Sukanya Nandyala and Haeng-Kon Kim. From cloud to fog and iot-based real-time u-healthcare monitoring for smart homes and hospitals. *International Journal of Smart Home*, 10:187–196, 02 2016.

[27] Andy Farnell. *Designing Sound*. The MIT Press, 2010.