SONOIDS: INTERACTIVE SOUND SYNTHESIS DRIVEN BY EMERGENT SOCIAL BEHAVIOUR IN THE SONIC DOMAIN

Andreas Apergis
Hellenic Open University,
Patras, Greece
std083780@ac.eap.gr

Maximos Kaliakatsos-Papakostas
Research and Innovation Center
“Athena”, Athens, Greece
maximos@ilsp.gr

Andreas Floros
Department of Audiovisual Arts,
Ionian University, Corfu, Greece
floros@ionio.gr

ABSTRACT

Artificial Intelligence (AI) algorithms have been extensively employed for generating art in many forms. From image generation to sound design, the self-emergence of structures in such algorithms make them suitable for exploring forms of computational creativity in automated generation of art. This work explores sound synthesis by combining swarm intelligence, user interaction and a novel sonic communication protocol between socially-capable artificial agents. These are termed “Sonoids” and socially behave following the well-known boids algorithm but perceive their environment (positions, velocities and identities of other agents) through sound. For the purposes of the work, the overall sound–synthesis environment is demonstrated as an iOS application that handles the Sonoids movement and sonifies their social interaction. User interaction is additionally allowed, which can modify specific parameters of the sonic communication protocol, leading to rich sonifications that present self-emergent structure.

1. INTRODUCTION

AI allowed the exploration of interesting blends between technology and art using interactive means. The self-emergence of structure in such algorithms incorporate random elements that preserve, however, intrinsic structure that potentially lead to interesting approaches in computational creativity. Specifically, well–known algorithms have been presented for the design of organized intelligent societies [1], which have also proven effective for optimization [2]. Among the most known implementations that simulate the shoaling and schooling behavior of autonomous agents is the “*Boids*” algorithm [1]. Many extensions of this model have been proposed that add social rules through which a more realistic behavior can be accomplished. For example, agents have been created that have been given the ability to avoid obstacles [3], have leadership skills [4], incorporate the artificial emotion of fear [5] and organize in predator–prey models [6] among many others [7]. Human perception of the swarm behavior has been studied in [8], providing indications about which human brain mechanisms may be involved in different behavioral aspects of the agents.

Humans identify elements of the social behavior of the boids algorithm in space. Therefore, this algorithm and its modifications have been employed for generating visual art. For example, in [9] swarms of Boids are evolved for transforming simple images in more complex forms, while in [10] the aforementioned framework was extended with additional chaotic dynamic features. The generated artwork exhibits highly complex dynamics, “inventing” a new glitch style. The visualization of sound using the boids algorithm has been examined in [11], where light sources are created according to the generated sound. Those light sources attract the Boid agents, the appearance of which is inspired by fireflies, resulting into interesting visualizations of musical pieces.

Several parts of the aforementioned social characteristics have been embodied to interactive agents, leading to music and sound output. A swarm that was able to improvise with symbolic music output with the guidance of a human singing voice was presented in [12]. The latter work was also enhanced with the addition of collision avoidance skills to the agents [13]. Symbolic music has also been composed by agents that were specialized to certain musical tasks [14]. A thorough review of these systems can be found in [15]. Such intelligent societies have been also used for granular synthesis [16, 17] extended with spatial characteristics too [18]. Finally, an interactive system has been proposed that receives feedback from the user to create audio and visual material using swarm intelligence and genetic algorithms [19]. More recently, the sonification of the Boids behavior has been integrated into the “Swarmlake” [20] game, which expanded the social behavior with user-controlled commands and attributed different agents with different sound properties, according to specific conditions of the game.

The paper at hand explores a creative sound–synthesis idea inspired by agents’ social interaction in the sonic domain. A novel sonic communication protocol is inherited to these agents, allowing them to communicate with sounds and to understand the locations, velocities and identities of neighboring agents. The targeted research hypothesis is whether interesting sound art can be produced by this approach that extends the original boids social behavior scheme mainly in terms of communicating and sensing: the social behavior that emerges through this communication and the social rules of the Boids algorithm. Additional simple and intuitive user interaction design allows the human user to control elements of this communication, to observe the resulting social behavior and to listen to the way that agents communicate under different communication parameters. For demonstrating the proposed sound–synthesis approach, a simulation software environment (iOS application) was realized, allowing for the derivation of sonic-content with variable and interesting form.

The rest of the paper is organized as following: Section 2 introduces Sonoids and analytically defines their collective behavior based on several functional extensions of the original Boids flocks. Next, Section 3 provides a thorough description of the developed simulation software application, outlining details on several parameters of the sound–synthesis control mechanism. Finally, Section 4 concludes the work and outlines potential future extensions of the Sonoids concept, targeted exclusively to the creative field of sound–art.

2. SONOIDS DESIGN

The main idea behind the Sonoids nature is the user interaction with an artificial community of autonomous agents that exhibit social behavior and communicate using sound signals. The social behavior part is borrowed from the *Boids* algorithm [1], while a novel sonic communication protocol is required for inheriting additional cognitive–related attributes to the agents for reproducing and perceiving sounds. Both these components are described in detail in the following paragraphs, starting from the boids concept.

2.1 Boids

Reynolds supplies Boids with the ability to act kinetically to the full extent of their surroundings, while at the same time restricting them to perceive only a part of this area, their neighborhood. With this constraint, the locality of the individual's behavior, which Reynolds attributes to the sensory perception of the environment primarily through vision, is functionally ensured. However, the Boids' model does not simulate a certain sense of the individuals (vision or hearing), but provides them with the information they would have extracted if they had been equipped with *perceptual and cognitive mechanisms*. This information consists of the position and velocity of every other agent in the environment and is made available to the individuals via a shared "database". Let $\vec{p_i}$ and $\vec{v_i}$ be the position and the velocity of Boid *i* in the current step of the algorithm, then in the next step its position is:

$$\vec{p_i}' = \vec{p_i} + \vec{v_i}, \tag{1}$$

and its velocity is given by the equation:

$$\vec{v_i}' = \vec{v_i} + \vec{a_i}, \tag{2}$$

where $\vec{a_i}$ is the centralized vector of the forces (acceleration) exerted on the specific Boid by the impact of coordinating behavior rules. The three main rules of coordination that Reynolds defines are the following:

1. Separation Rule: to avoid conflicts with its neighboring individuals.
2. Alignment Rule: to comply with the mean orientation of its neighboring individuals.
3. Coherence Rule: to comply with the mean spatial distribution of its neighboring individuals.

The contribution of each rule (i.e. are the acceleration vectors $a\vec{s}_i$, $a\vec{a}_i$, $a\vec{c}_i$ respectively) can be combined as:

$$\vec{a_i} = a\vec{s}_i + a\vec{a}_i + a\vec{c}_i \tag{3}$$

forming the above centralized vector of the forces $\vec{a_i}$.

If d_{min} is the minimum allowed distance between two Boids (smaller distances trigger separation), d_{nei} is the radius of a Boid's perceived neighborhood and $j \in C$, where C is the set of N Boids that satisfy the condition $\|\vec{p_i} - \vec{p_j}\| < d_{min}$ for the first rule or the condition $\|\vec{p_i} - \vec{p_j}\| < d_{nei}$ for the second and third rules, with $j \neq i$, then

$$v\vec{s}_i = \frac{1}{N} \sum_{\forall j \in C} \vec{p_i} - \vec{p_j}, \tag{4}$$

$$v\vec{a}_i = \frac{1}{N} \sum_{\forall j \in C} \vec{v_j}, \tag{5}$$

$$v\vec{c}_i = \frac{1}{N} \sum_{\forall j \in C} \vec{p_j}. \tag{6}$$

2.2 Sonoids

As it was mentioned before, Sonoids are intended to exhibit similar collective behavior with Boids. They must initially be equipped with the ability to act kinetically throughout the free space of their environment. Therefore Sonoids inherit from Boids the properties related with the instantaneous position and the position update. In addition, they must confine themselves to the perception of their neighborhood, as Boids do, within which they comply with the three rules of coordination. Sonoids' primary differentiation from Boids is their way of perceiving the environment, which is still sensory but based solely on the sense of hearing. Towards this functionality, it is necessary to equip Sonoids with the following three mechanisms:

- Signaling Mechanism: emit an audio signal in order to be perceived by the neighboring individuals.
- Perceptual Mechanism: receive and analyze audio signals in order to perceive the neighboring individuals.
- Cognitive Mechanism: for processing of the analyzed audio signal and extracting location, orientation, and speed information of neighboring individuals.

The particularity of the Sonoids model lies in their different ways of accessing the shared "database" per Mechanism in operation. While they are given the right to retrieve location data of other Sonoids in order to consider them neighbors during the operation of the Perceptual Mechanism, this right is removed from them when the Cognitive

Mechanism is put into operation. In this way the newly introduced signal data in the model is adapted to the metrics of the already existing geometric position; the signal determines the update of the position and extends the original model. For that, the relationships that connect the intensity I of the signal to its amplitude A (eq. 7) and the spatial distance d from the source (eq. 8) will be used:

$$I \propto A^2, \tag{7}$$

$$I \propto \frac{1}{d^2}. \tag{8}$$

2.2.1 *Signaling Mechanism*

The parameterization rules of the signal that the Sonoid i emits through this Mechanism define the protocol of Sonoids' communication as follows:

- Form: sinusoidal $x_i(n) = A_0 \cos(\omega_i n T_s)$ so that its frequency spectrum consists of one frequency, where T_s is the sampling period of the signal.
- Frequency: Distinct and unique ω_i for each Sonoid so that it is distinguished by others that might potentially be within range at the same time.
- Amplitude: A_0 constant in time, so that the exact location of its source can be extracted.
- Emission Point: the Sonoid's center of mass $\vec{p_i}$ with omnidirectional scattering.

2.2.2 *Perceptual Mechanism*

The Perceptual Mechanism consists of four receptors that are uniformly distributed on the perimeter of the Sonoid's i body at the following points (see also Fig. 1):

$$\vec{p_{iFront}} = (r \cdot \cos(\theta) + x, r \cdot \sin(\theta) + y) \tag{9}$$

$$\vec{p_{iRight}} = \left(r \cdot \cos\left(\theta + \frac{\pi}{2}\right) + x, r \cdot \sin\left(\theta + \frac{\pi}{2}\right) + y\right) \tag{10}$$

$$\vec{p_{iRear}} = (r \cdot \cos(\theta + \pi) + x, r \cdot \sin(\theta + \pi) + y) \tag{11}$$

$$\vec{p_{iLeft}} = \left(r \cdot \cos\left(\theta + \frac{3\pi}{2}\right) + x, r \cdot \sin\left(\theta + \frac{3\pi}{2}\right) + y\right), \tag{12}$$

where it is assumed that the Sonoid i is located at $\vec{p_i} = (x, y)$, its velocity is $\vec{v_i} = (x', y')$, $\theta = arctan\left(\frac{x'}{y'}\right)$ and its body extends at a distance r from the center of its mass. Transferring the model into three dimensions is trivial as that would only require six receptors.

The Perceptual Mechanism operates in two phases: one for the collection of signals and the other for their analysis. During the signals collection phase, for $j \in C$, where C is the set of sonoids that satisfy the condition $\|\vec{p_i} - \vec{p_j}\| < d_{nei}$ and emit signal $x_j(n)$, the Sonoid's i receptor (one of the four) which is located at the point $\vec{p_{iS}}$ collects the signal:

$$x_{iS}(n) = A_{max}(d_{min} - r) \sum_{\forall j \in C} \frac{x_j(n)}{\|\vec{p_{iS}} - \vec{p_j}\|}, \tag{13}$$

where A_{max} is a predetermined value for the amplitude of the signal that a Sonoid receives from another when the

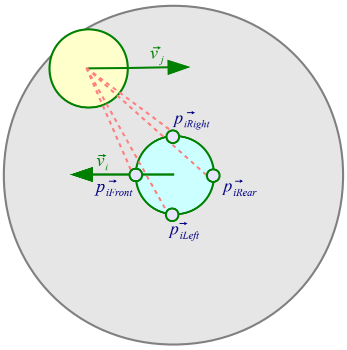

Figure 1. Sonoid's Receptors.

distance between their centers of mass is equal to d_{min}. Specifically, the value A_{max} corresponds to the distance $(d_{min} - r)$ since the receptor is located on the perimeter of the Sonoid's body and the amplitude takes that value when the points $\vec{p_j}$, $\vec{p_{iS}}$ and $\vec{p_i}$ are collinear as shown in Fig. 2. Equations (7) and (8) are used in equation (13) for the amplitude scaling of the emitted signal from $\vec{p_j}$ while being received at $\vec{p_{iS}}$.

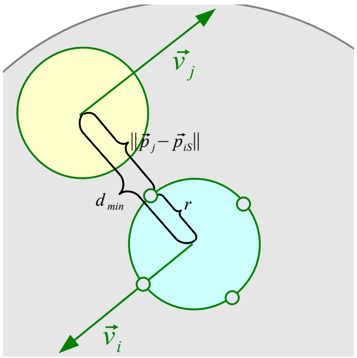

Figure 2. Minimum Allowed Distance.

During the analysis phase, the signal collected at each receptor is transformed from the time domain to the frequency domain using the Discrete Fourier Transform. For the collected signal $x_{iS}(n)$ of the receptor at $\vec{p_{iS}}$ the spectral sample at frequency ω_k would be:

$$X_{iS}(\omega_k) = \sum_{n=0}^{N-1} x_{iS}(n) e^{-j\omega_k n T_s} \tag{14}$$

where N is the length of the signal and $k = 0, 1, 2, \ldots, N-1$.

2.2.3 *Cognitive Mechanism*

Through the Cognitive Mechanism, Sonoids acquire the skills of further processing the analyzed signals to extract location information of their sources. Since the sources are other Sonoids that coexist in the environment, each Sonoid possesses as prior knowledge the distinct frequencies of the emitted signals. If ω_i is the frequency of the signal emitted by Sonoid i, according to the pre-agreed protocol:

- if the frequency stays constant at all steps in the algorithm, Sonoid knows from the beginning all distinct frequencies ω_j for $j \neq i$.
- if the frequency stays constant at any step in the algorithm, the specific Sonoid knows how the frequency of the signal that it emits is related to the frequencies of the signals that the other Sonoids emit. This ensures that the proportions between frequencies remain constant throughout the length of the algorithm, i.e. $\omega_i = m_j \cdot \omega_j$ for $j \neq i$ and $m_j \in \Re$.

The L-largest in magnitude spectral samples are selected (with $1 \leq L \leq N-1$) and tested for frequency matching in the set of Sonoid's prior knowledge. Absolute matching takes place only for the ω_j frequencies that are integer multiples of $\Omega = 2\pi f_s/N$, where f_s is the sampling frequency in Hz. For all others, matching is achieved if ω_j is within the range of $(\omega_{k-1/2}, \omega_{k-1/2}]$, where ω_k is the central frequency of the spectral sample. In a possible extension of the Sonoids implementation in the physical world, this margin would be also necessary for taking under consideration the frequency deviations introduced by the Doppler effect. Since all agents are normally in motion relative to each other, when agents approach each other their frequencies will tend to be perceived as higher for each Sonoid (and lower when Sonoids move away from each other); the margin would therefore be necessary to compensate for those deviations. From the above, an additional parameterization rule of the Signaling Mechanism seems to be necessary: *consecutive signal frequencies must differ by at least* Ω, the value of which is defined by the maximum possible speed that the Sonoids can reach – and the consequent maximum possible frequency modulation by the Doppler effect. When the matches are completed, each Sonoid knows "who" are its neighbors.

For finding "where" its neighbors are, each Sonoid needs the data of their distance and angle, as computed by its receptors. Starting with the estimation of angle, for the receptor located at $\vec{p_{iS}}$ whose input signal contains the identified frequency ω_j the following vector is calculated:

$$\vec{q_{iS}} = |X_{iS}(\omega_l)| \frac{\vec{p_{iS}} - \vec{p_i}}{\|\vec{p_{iS}} - \vec{p_i}\|}, \tag{15}$$

where $|X_{iS}(\omega_l)|$ is the magnitude of the spectral sample at frequency ω_l which had been previously matched with ω_j. The sum of the vectors for all the receptors whose input signals contain that frequency is the vector $\vec{q_{i,j}}$ that starts from Sonoid's i center of mass and points to the neighboring Sonoid j (see Fig. 3 for details):

$$\vec{q_{i,j}} = \vec{q_{iFront}} + \vec{q_{iRight}} + \vec{q_{iRear}} + \vec{q_{iLeft}}. \tag{16}$$

For the estimation of the distance, the problem comes down to resolving a side-side angle (SSA) triangle which, as is well known, might not have a solution (ambiguous case). However, here we have the luxury of having four triangles, one for each Sonoid's receptor. So starting from the front receptor we can go through all four of them until we find a solution. Below, the equations 17-21 refer to the front receptor and can be explained by observing Fig. 4.

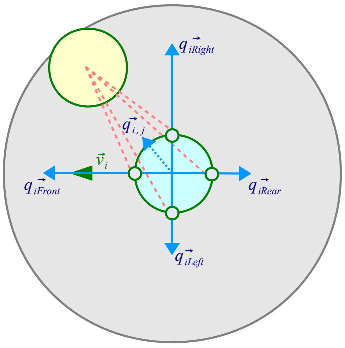

Figure 3. Estimation of Angle.

The angle $\theta_{iFront,j}$ of the perceived position of Sonoid j with the front receptor of Sonoid i from the center of mass of the latter is that of:

$$\theta_{iFront,j} = arccos\left(\frac{\langle \vec{q_{i,j}}, \vec{q_{iFront}} \rangle}{\|\vec{q_{i,j}}\| \cdot \|\vec{q_{iFront}}\|}\right) \tag{17}$$

The angle $\theta_{iFront,i}$ of Sonoid's i center of mass with its front receptor from the perceived position of Sonoid j is provided from:

$$\theta_{iFront,i} = arcsin\left(\frac{2|X_{iFront}(\omega_l)|}{A_{max}(d_{min} - r)} r \sin(\theta_{iFront,j})\right) \tag{18}$$

where equations (7) and (8) were considered again.

The angle $\theta_{i,j}$ of the perceived position of Sonoid j with the Sonoid's i center of mass from the front receptor of the latter is given by:

$$\theta_{i,j} = \pi - \theta_{iFront,j} - \theta_{iFront,i} \tag{19}$$

Then, the distance $d_{i,j}$ of the perceived position of Sonoid j from Sonoid's i center of mass is calculated as:

$$d_{i,j} = \frac{r \sin(\theta_{i,j})}{\sin(\theta_{iFront,i})} \tag{20}$$

and the position of Sonoid j as perceived by Sonoid i is:

$$\vec{p_{j,i}} = \vec{p_i} + d_{i,j} \frac{\vec{q_{i,j}}}{\|\vec{q_{i,j}}\|} \tag{21}$$

It is obvious that at least two consecutive steps of tracking a Sonoid are required by another one to estimate the velocity of the former, so the latter is not able to comply with the Alignment Rule during the first step.

2.3 Combined use in a musical context

Assuming that the minimum desired outcome of the Sonoids algorithm is to demonstrate a similar to the Boids emerging behavior, which is audibly observable and aesthetically interesting, a sufficiently large audible "space" for exploration would be required. This "space" consists of signals other than the operational signals of the algorithm (otherwise that "space" would be strictly parametrized according to the rules of the Signaling Mechanism), but the output of

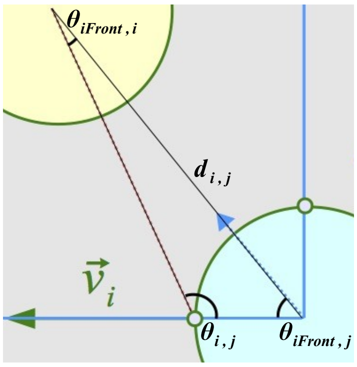

Figure 4. Estimation of Distance.

the latter allows the user to explore the "space" by controlling the parameters of the resulting signals. The user should evaluate the output of the algorithm based on the achieved audio effect.

Another evaluation of the output takes place inside the system before it is sonified in order to further expand the audible "space": since it is possible to execute the Boid algorithm in parallel with the Sonoids one, at each step the distance between the outputs of the latter and the former can be evaluated. In other words, during this evaluation, we aim to address the question "how a Sonoid qualifies as a Boid". By setting appropriate weights in the position and orientation deviations between a Sonoid and the corresponding Boid, new data are obtained to contribute to the final audio effect ultimately evaluated by the user. Instead of using symbolic musical data, timbre was considered to be the most appropriate of the basic characteristics of the musical sound for the parallelization of the latter with the modeled swarm. As the overall emerging behavior of the swarm follows the local interactions of the entities that comprise it, the timbre of a musical sound can be considered as the overall effect of the influence of its partial frequencies on its perceived identity.

Another significant aspect that should be mentioned is that the Sonoids model holds an advantage over the corresponding model of Boids for use in a musical context because of its acoustic awareness. While a Boid acts based on the actual position of its neighbors, a Sonoid acts based on the position it extracts through the intensity of the signals emitted by its neighbors to whom a fixed or temporary discrete frequency has been assigned. Recalling the issue of distinguishing the operational signals of the algorithm from the signals that synthesize the sound output, and focusing on musical sound, the proposed parameters set for each signal type is:

- Operational Signals: Their frequency values are constant and form a harmonic series that coincides with the central frequencies appeared during their spectral analysis. This promotes DFT for its mathematical purity and not for possible "glitch" aesthetics produced by spectral leakage or ways to overcome it. Their instantaneous width values are set by the user.
- Output Sound Signals: Of sinusoidal form and mixed together by means of additive synthesis (Equation 22). The user selects tonal pitch and volume by setting the initial frequency and amplitude values of the signal that corresponds to the first partial or base frequency. The momentary frequency and amplitude values of a signal are set by an individual agent of the swarm; hence, the number of signals is equal to the number of individuals.

According to the above proposals, the i-th Sonoid represents the i-th harmonic and controls the i-th partial frequency of the sound that the user receives. The user controls the intensities of the signals that Sonoids emit with that possibly resulting into the Sonoids' distorted judgment about the location of their neighbors. Each side partially controls the audible signals of the other. The intensity of the i-th partial frequency is adjusted based on the orientation and distance of the i-th Sonoid (i-th harmonic) relative to the individual representing the 1st harmonic (equations 23-25). Finally, the value of the i-th partial frequency is adjusted based on the position deviation of the i-th Sonoid (i-th harmonic) from what it would had if it was a Boid (equations 26-27).

$$output(n) = \sum_{k=1}^{K} A_k \cdot \cos(\omega_k n T_s) \tag{22}$$

$$A_k = \frac{1}{K} \cdot A_{base} \cdot O_k \cdot D_k \tag{23}$$

$$O_k = 1 - \frac{|\angle(\vec{v_k}, \vec{v_1})|}{\pi} \tag{24}$$

$$D_k = 1 - \frac{\|\vec{p_k} - \vec{p_1}\|}{d} \tag{25}$$

$$\omega_k = k \cdot \omega_{base} \cdot E_k \tag{26}$$

$$E_k = 1 \pm \frac{\|p_{k(\vec{Sonoid})} - p_{k(\vec{Boid})}\|}{d'} \tag{27}$$

where A_{base} and ω_{base} are respectively the amplitude and frequency values of the signal that corresponds to the base frequency, while d and d' are some distance-informed coefficients.

3. SIMULATION IMPLEMENTATION

A system based on the proposed Sonoids' model was initially prototyped using the ChucK [21] programming language for desktop environment. After this initial implementation step, it was also realized as a software application for the widely-employed iOS mobile operating system. The "backbone" of the iOS realization was based on openFrameworks [1], while the Digital Signal Processing (DSP) part was realized using two additional programming frameworks: for the synthesis and analysis of the Sonoids' signals, Apple's Accelerate framework [2] was employed in

---

[1] `http://openframeworks.cc`

[2] `https://developer.apple.com/documentation/accelerate`

order to exploit the parallel computations through the use of Single instruction, Multiple Data (SIMD) instructions to the mobile hardware. For the synthesis of the audio that the application generates, Faust2api [22] was used not only for the heavily-optimized DSP code the Faust compiler produces, but also for the convenience that this Application Protocol Interface (API) provides for handling polyphony.

The simulation employed a modeled Sonoids swarm of 80-120 entities, with a refresh rate of 30 frames per second. It was executed on a five years old iOS hardware platform. The number of entities determine the size of the DFT used by each one of them, so that the size of its corresponding FFT must be a power of 2 greater than two times the number of entities. Further ways for optimizing the Boids' algorithm that could be also applied to the Sonoids' case (such as bin-lattice spatial subdivision [23]) were not adopted, since they realistically benefit implementations of significantly more than 100 entities, while the above FFT requirement is the real "bottleneck" for the performance of our system.

In the following subsections we present implementation details of the real-time control of the duration, pitch and volume of the sound output, as well as the real-time yet indirect control of its timbre through the modification of the Sonoids' signals.

3.1 Controlling the Duration of Generated Sound

The members of the swarm may be in one of two general modes regarding signal production, either in a sound-generation or in a silent mode. On the other hand, they constantly remain in the listening mode. The user touches the hand held device screen and sets the swarm in sound-generation mode. He/she then hears the synthesized sound. Otherwise, the swarm is set into silent mode where the user does not hear any sound. At the same time, when the swarm of Sonoids is in the sound-generation mode, its members are provided with the ability to extract the location of their neighbors in order to comply with the rules of coordinating behavior and develop a similar to the Boids organization (Fig. 5 right). Otherwise when the swarm is silent, its members retain the speed and orientation they had just before this mode changed and do not develop any organization (see Fig. 5 left).

The above convention of controlling the duration of sound was adopted in order to make clear and intuitive to the user that the sound he/she hears originates from the observed swarm being emerged on the screen. In the right part of Fig. 5, the black outlined circle corresponds to the user's contact point with the screen, while the lines starting from an entity end up to the position of its considered neighbors. The transparent "images" of the entities that appear mostly on the left part of Fig. 5 are the positions that the Sonoids of the same color should have if they were Boids.

3.2 Controlling the Pitch and Volume of Generated Sound

The user selects the pitch and volume of the sound he/she hears through the exact touching point on the hand held device screen: the x-axis determines pitch and the y-axis the volume (see Fig. 6). User's choice is a general suggestion for the above characteristics of the generated sound by setting the initial frequency and amplitude values of the signal corresponding to the first partial (or base) frequency, while the members of the swarm undertake to instantly set these values for all the partials as proposed in subsection 2.3 for as long the user touches the screen and the swarm stays in sound-generation mode.

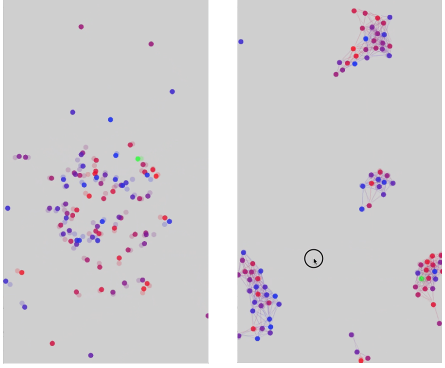

Figure 5. Silent Mode (left) and Sound Mode (right).

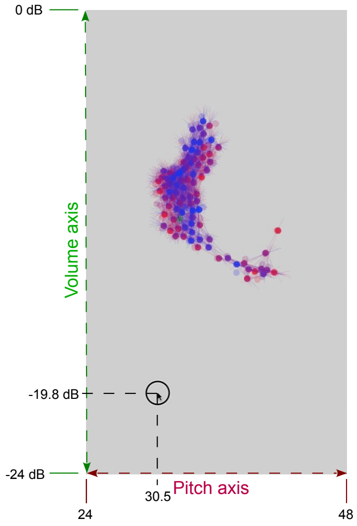

Figure 6. Pitch (in MIDI note numbering) and Volume (in dB) axes.

3.3 Controlling the Amplitude of Sonoids' Signals

When the swarm is in sound-generation mode, the user can specify the amplitude of the signals emitted by the Sonoids. The instantaneous amplitude value is common for all members of the swarm. It is determined by the user's point of contact with the screen along the axis of the dimension that is also responsible for determining the generated sound volume. This two signal types amplitude coupling is employed in order to effectively simplify the performed user actions.

When the user touches a point on the hand held device screen that corresponds to a level on the amplitude scale which is different from a pre–determined level that ensures the secure estimation of its neighbors' real positions, an even more dramatic change in the timbre of the generated sound takes place. This pre–determined amplitude value is set at the middle range of the amplitude scale. In this case, the derived position of the Sonoid neighbors is very close to the real, due to the lack of "images" of the entities illustrated in Fig. 7a. This fact results into the generation of a semi-periodic sound signal as indicated by the frequency spectrum of Fig. 8a. On the other hand, amplitude values at the lower or higher range of the amplitude scale lead to the occurrence of non-periodicity in the output sound signal. In particular, for the lower levels of the scale, the entities overestimate the distance from their neighbors and tend to accumulate at a distance less than the minimum allowed (see Fig. 7b). On the contrary, for higher scaling levels, the Sonoids underestimate the distance between them, and they are repelled at high speed since they consider this distance shorter than the minimum allowed as it is illustrated in Fig. 7c. The spectrum for the snapshots of Figures 7b and 7c (where the "images" of the entities are apparent) is shown in Figures 8b and 8c respectively and clearly indicate the non–periodicity of the generated sonic content.

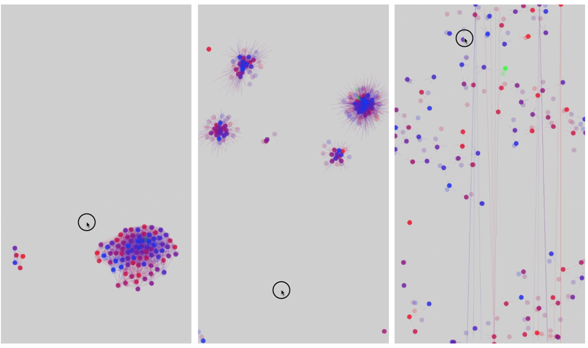

Figure 7. Levels of Amplitude Scale: (from left to right) (a) Middle, (b) Low, (c) High

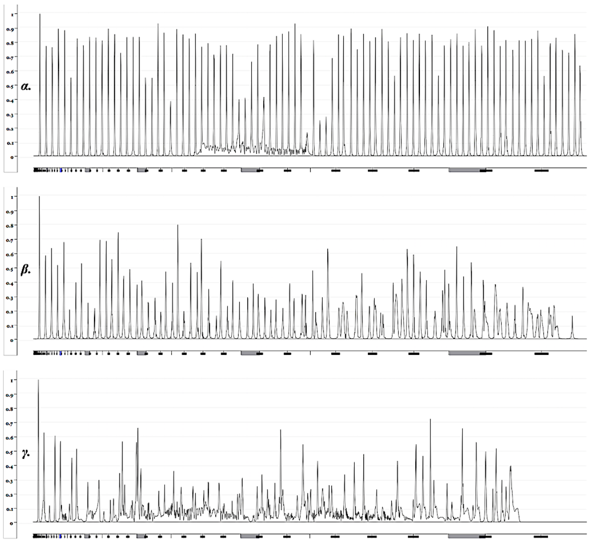

Figure 8. Frequency Spectra of Generated Sound during the Snapshots of Fig. 7: (from top to bottom) (a) Fig. 7a, (b) Fig. 7b, (c) Fig. 7c

4. CONCLUSIONS

In this paper we have introduced a new sound synthesis approach based on Sonoids, an extension of the original socially-capable artificial Boids. Compared to Boids, Sonoids communicate and perceive the hosting swarm environment through sound signals that are reproduced by the artificial individuals. Towards this aim, a novel sonic communication protocol has been developed that allows the Sonoids to effectively perceive their environment, i.e. the positions, velocities and identities of other swarm agents. Their overall social interaction is based on simple rules of common Boids and is sonified using the combination of an advanced control mechanism that is able to derive complex electronic music content with intuitive user interaction that supplementary controls elements of the Sonoids auditory communication. For the purposes of this work, the Sonoids sound synthesis framework has been prototyped as an iOS software application that allows the user to experiment with different parameters of the sonic communication protocol in a fully simulated environment; user interactions lead to different sound output and, consequently, to different formations of social behavior. The final result is the generation of sonic content that is evolved according to the observed movement of the agents. A video demonstrating a Sonoids interaction session can be accessed here [3] .

Immediate future extensions of this work may include additional visualization and sonification options, e.g. the illustration of a real–time spectrogram, or the option to employ different sound generators for each agent. Specifically, there is intention of augmenting the audio engine with components suggested from practices of physical modeling synthesis since we aim to further parallelize the physically-informed aspect of the algorithm with that of sound. On the other hand, long–term future work may investigate hardware (i.e. "non–simulated") implementation approaches of the Sonoids, either with ground–roaming robots or quadrotors. Research in the field of robotics has shown that the Boids algorithm can be employed for collision detection in quadrotors [24]. Not only it would be interesting to observe the behavior of the Sonoids in real-world applications but also to explore the spatial characteristics of the produced sonic scape in such environments.

5. REFERENCES

[1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Net-*

---

[3] `https://www.youtube.com/watch?v=clVktKw_yvk`

works, 1995. Proceedings, vol. 4. IEEE, Dec. 1995, pp. 1942–1948 vol.4.

[3] C. W. Reynolds, “Not bumping into things,” in *in the notes for the SIGGRAPH 88 course Developments in Physically-Based Modeling*, ser. ACM SIGGRAPH, 1988, pp. G1–G13.

[4] C. Hartman and B. Benes, “Autonomous boids: Research articles,” *Comput. Animat. Virtual Worlds*, vol. 17, no. 3-4, pp. 199–206, Jul. 2006.

[5] C. Delgado-Mata, J. I. Martinez, S. Bee, R. Ruiz-Rodarte, and R. Aylett, “On the use of virtual animals with artificial fear in virtual environments,” *New Gen. Comput.*, vol. 25, no. 2, pp. 145–169, Nov. 2007.

[6] R. Charrier, C. Bourjot, and F. Charpillet, “Modeling swarm phenomena using logistic agents: application to a predators-prey pursuit,” in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, ser. AAMAS ’09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 1245–1246.

[7] C. K. Hemelrijk and H. Hildenbrandt, “Some causes of the variable shape of flocks of birds,” *PLoS ONE*, vol. 6, no. 8, p. e22479, 08 2011.

[8] J. Harvey, K. E. Merrick, and H. A. Abbass, “Assessing human judgement of computationally generated swarming behaviour,” *Frontiers in Robotics and AI*, vol. 5, p. 13, 2018.

[9] T. J. Choi and C. W. Ahn, “A swarm art based on evolvable boids with genetic programming,” *Journal of Advances in Information Technology Vol*, vol. 8, no. 1, 2017.

[10] ——, “Artificial life based on boids model and evolutionary chaotic neural networks for creating artworks,” *Swarm and Evolutionary Computation*, 2017.

[11] A. Rodrigues, P. Machado, P. Martins, and A. Cardoso, “Sound visualization through a swarm of fireflies,” in *Portuguese Conference on Artificial Intelligence*. Springer, 2015, pp. 664–670.

[12] T. M. Blackwell and P. Bentley, “Improvised music with swarms,” in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC ’02*, vol. 2. IEEE, 2002, pp. 1462–1467.

[13] ——, “Don’t push me! collision-avoiding swarms,” in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC ’02*, vol. 2. IEEE, 2002, pp. 1691–1696.

[14] T. M. Blackwell, “Swarm Music: improvised music with multi-swarms,” in *Proceedings of the 2003 AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, 2003, pp. 41–49.

[15] ——, “Swarming and music,” in *Evolutionary Computer Music*, E. R. Miranda and J. A. Biles, Eds. Springer London, 2007, pp. 194–217.

[16] T. Blackwell and M. Young, “Swarm Granulator,” in *Applications of Evolutionary Computing, Proceedings of the 2nd European Workshop on Evolutionary Music and Art, EVOMUSART 2004*. Springer Berlin Heidelberg, 2004, pp. 399–408.

[17] T. Blackwell, “Swarm granulation,” in *The Art of Artificial Evolution*, ser. Natural Computing Series, J. Romero and P. Machado, Eds. Springer Berlin Heidelberg, 2008, pp. 103–122.

[18] S. Wilson, “Spatial Swarm Granulation,” *Proceedings of the 2008 International Computer Music Conference*, Aug. 2008.

[19] D. Jones, “Atomswarm: a framework for swarm improvisation,” in *Proceedings of the 2008 conference on Applications of evolutionary computing*, ser. Evo’08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 423–432.

[20] M. Kaliakatsos-Papakostas, A. Floros, K. Drossos, K. Koukoudis, M. Kyzalas, and A. Kalantzis, “Swarm lake: A game of swarm intelligence, human interaction and collaborative music composition,” in *ICMC*, 2014.

[21] G. Wang, P. R. Cook *et al.*, “Chuck: A concurrent, on-the-fly, audio programming language.” in *ICMC*, 2003.

[22] R. Michon, J. Smith, C. Chafe, S. Letz, and Y. Orlarey, “faust2api: A comprehensive api generator for android and ios,” in *Proceedings of the Linux Audio Conference (LAC-17), Saint-Etienne, France*, 2017, pp. 18–21.

[23] C. W. Reynolds, “Interaction with groups of autonomous characters,” in *Game developers conference*, vol. 2000, 2000, p. 83.

[24] R. G. Braga, R. C. da Silva, A. C. Ramos, and F. Mora-Camino, “Collision avoidance based on reynolds rules: A case study using quadrotors,” in *Information Technology-New Generations*. Springer, 2018, pp. 773–780.