TouchNoise: A Particle-based Multitouch Noise Modulation Interface

Axel Berndt
Chair for Media Design and Interactive Media Lab
Technische Universität Dresden
Dresden, Germany
Axel.Berndt@tu-dresden.de

Nadia Al-Kassab
Technische Universität Dresden
Dresden, Germany
Nadia.Al-Kassab@mailbox.tu-dresden.de

Raimund Dachselt
Interactive Media Lab
Technische Universität Dresden
Dresden, Germany
Raimund.Dachselt@tu-dresden.de

ABSTRACT

We present the digital musical instrument *TouchNoise* that is based on multitouch interaction with a particle system. It implements a novel interface concept for modulating noise spectra. Each particle represents a sine oscillator that moves through the two-dimensional frequency and stereo panning domain via Brownian motion. Its behavior can be affected by multitouch gestures allowing the shaping of the resulting sound in many different ways. Particles can be dragged, attracted, repelled, accentuated, and their autonomous behavior can be manipulated. In this paper we introduce the concepts behind this instrument, describe its implementation and discuss the sonic design space emerging from it.

Keywords

digital musical instrument, multitouch

1. ACTIVE MUSICAL INSTRUMENTS AND MULTITOUCH

Since the first arpeggiators, electronic musical instruments developed a more and more autonomous behavior. Complex cascades of sound events no longer have to be triggered sound by sound but whole sequences at once. Growing computational power and the development of versatile digital musical instruments lead to a new quality of autonomous behavior and to what Rubén Hinojosa Chapel defines an *active musical instrument*: "The system actively proposes musical material in real-time, while the user's actions [...] influence this ongoing musical output rather than have the task to initiate each sound." [3] One of the first instruments of this kind is Joel Chadabe's "Interactive Music Composition and Performance System" giving the user high-level control over the realtime music generation [2].

Controlling such behavior is no trivial task and puts high requirements on the design of user interfaces. Traditional musicians are used to a very direct and physical interaction with their instrument's sound generation. Typical examples are keystroke velocity, bowing and lip pressure which modulate sound parameters such as loudness, timbre, and pitch. The control of an active musical instrument, in contrast, takes place on a higher level of abstraction and addresses the instrument's behavior while autonomously generating sequences of sounds. Basically, such interaction edits parameters of algorithms. In the worst case this may mean to edit numeric fields in a digital form, an admittedly extreme example. Traditional musicians often feel uncomfortable with this loss of direct, embodied control.

During the past decade, research in computer music and digital musical interfaces utilized technological advances in human-computer interaction, such as motion tracking, multitouch technologies, and tangible user interfaces, to bring back physicality and directness into musical interaction. To name just a few examples: Miha Ciglar's [4] ultrasound-based instrument gives tactile feedback and tracks hand gestures. Hansen and Jensenius [10] use sensor-equipped pants as drum interface. Gelineck & Böttcher [9] present a framework called *6to6Mappr* that interfaces several devices, including Microsoft's Kinect, Nintendo's Wiimote, accelerometers, and touch input.

Among all the different modalities, touch input is the one that is today probably most common and frequently used for controlling generative music systems and active musical instruments, as indicated by the variety of apps for smartphones and tablets. Well known are the ambient music apps by Eno and Chilvers [6, 7]. These represent musical material by geometric forms which are arranged on the 2d plane. Gelineck et al. exploit the advantages of multitouch interaction for music mixing [8]. Well known is also the *reacTable* [12] for its pioneering musical interface that combines multitouch and tangible interaction. *SpaceWiz* by Rudess and Miller [13] simulates particles (asteroids) that collide with moving objects (planets) and, thereby, create ever-changing musical patterns.

Particle systems, multi-agent systems, and systems with self-regulating behavior can often be found as central components of active musical instruments. An approach by Eldridge [5] creates musical structures from a homeostatic system that is excited by user interaction. The musical game *Electroplankton* [11] offers several playing modes; one of them, named Luminaria, is a melodic progression graph that is traversed by up to four sound generators (plaktons) while the user edits the graph. The Android app *Orbits* [16] simulates particle motion that is influenced by attractive forces of other particles nearby, and by the particle's size which maps to pitch. *NodeBeat* [14] demonstrates a clever combination of pulse generators and sound generators visually represented by nodes in a graph; proximity to other nodes creates connections. The graph forms a sequencer that is affected by the nodes' movement. Blackwell [1] and Unemi & Bisig [15] investigate swarming algorithms as basis of interactive generative music processes.

The previous examples demonstrate how manifold and

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

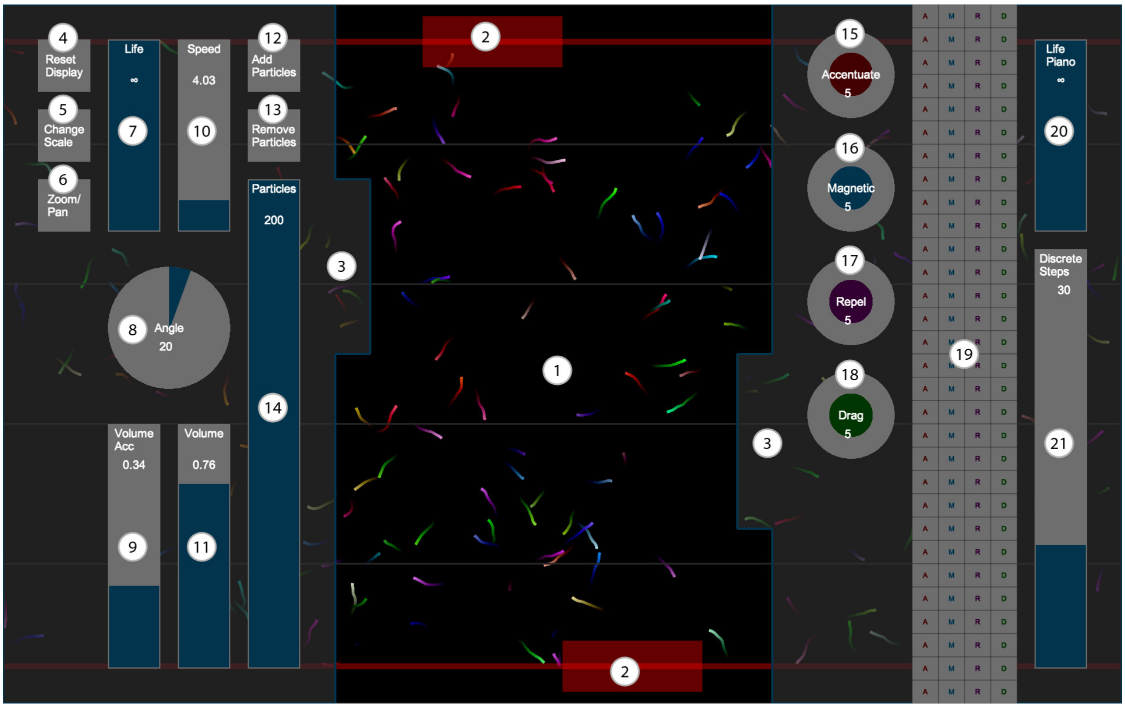


Figure 1: The user interface of *TouchNoise*. Detailed descriptions are given in Section 2.

important the visual concepts of such interfaces are. Well-designed interfaces feature an inherent connection of the visual phenomena to the musical structures and sonic output that deduce from them. Regarding active musical instruments, the visual channel is used to express the instrument's autonomous behavior and to create affordances to interact with audiovisual elements. Touch interaction enables direct manipulation of these visual elements and, thus, the sonic structures they represent. This regains some directness and physicality to the interaction that is so important to musicians. The visual qualities are often not just interface metaphors for the musicians but also used for overhead projection in live concert situations.

With our active musical instrument *TouchNoise*, we address the same issues and came to an innovative approach that is an interesting contribution to the field in several aspects. Noise spectra are created and modulated via multitouch interaction with a system of autonomous sound particles. This interaction turns out to be very direct and intuitive due to our mapping approach of visual (multitouch interactive) phenomena to sonic behavior. Different interaction modes and their combinations open up a wide range of playing techniques and give access to a number of musically interesting sonic phenomena.

After these introductory lines we introduce the concept of *TouchNoise* in Section 2. The sonic design space and typical playing techniques are described in Section 3. A critical reflection in Section 4 motivates future developments followed by Section 5 which summarizes this paper.

2. THE INSTRUMENT

The basic concept of the digital musical instrument *TouchNoise* that we describe in the remainder of this paper is, simply put, as follows: Sound particles, each creating a sine tone while autonomously moving in the frequency and stereo panning domain, are visually represented as points on the 2d plane (stereo position on the x-axis, frequency on the y-axis). The movement is based on Brownian motion. The particles can be affected by multitouch interaction in many different ways (e.g., drag, attract, repel etc.). All sine oscillators are added and create noise spectra with different density regions and sonic agility controlled by the user. A screenshot of the user interface is shown in Figure 1.

2.1 User Interface

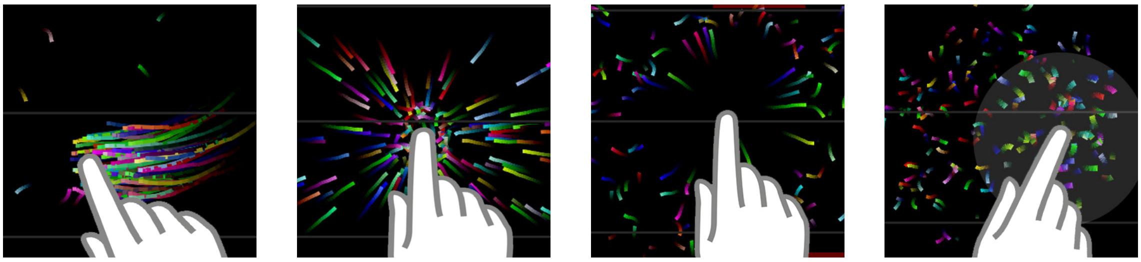
The general layout of the user interface is organized in three sections. The central section is the *particle playground* (no. 1 in Fig. 1). This is the domain of the sound particles where any direct interaction with the particle system takes place. The horizontal axis maps onto stereo panning, the vertical axis maps onto frequency. The particles are visually represented on the playground at the position that corresponds to the sine tone they output.

Two drawers (no. 3 in Fig. 1), one to the left and the other to the right, contain the other two sections. The left drawer contains global settings of the particle system, such as particle creation, lifetime, motion characteristics, and volume levels. The right drawer contains mode switches and settings for direct interaction on the particle playground.

2.2 Particle Behavior and Interaction

Particles are created in two ways. Raising the particle slider (no. 14 in Fig. 1) creates particles at random positions. The *add* mode (no. 12) creates particles only when the user taps into the particle playground. The particles appear at the position of the tap. They immediately output their corresponding sine tone with frequency and stereo position according to their position on the playground. Removing particles is likewise possible either by lowering the particle slider to remove randomly chosen particles or by tapping at the particles to be removed (*remove* mode, no. 13).

All particles start with a random initial direction. Their further movement is based on Brownian motion. Stereo position and frequency modulate accordingly. The step size is set by the speed slider (no. 10). The higher its value, the greater the step size, resulting in faster particle motion and brisker sonic output. The maximum angle of the random rotation is limited by the angle ruler (no. 8). A minimum



(a) Drag mode. (b) Magnetic mode. (c) Repel mode. (d) Accentuate mode.

Figure 2: Modes of direct interaction with the particles.

setting of 0° creates linear motion, whereas the maximum of 360° allows the full positive and negative rotation. With such great ranges the particles tend to stick around their current position which creates a relatively steady sound. The frequency range of the particles can be limited by an upper and a lower border (no. 2) with lowest and highest settings of 20Hz and 20kHz.

Furthermore, a lifetime slider (no. 7) allows the user to limit the particles' presence after creation. Possible settings reach from 100 milliseconds up to 1 minute. The lifetime boundary can also be deactivated. In this case, particles are present until they are removed by interaction.

Beyond these general behavioral settings, several modes for direct interaction within the particle playground are implemented. *Drag* mode (no. 18 and Fig. 2(a)) may be the most obvious: When tapping into the playground each particle that gets under the tap will be held by it and can be dragged. The more particles are collected this way, the more prominent is the sonic output of that cluster. Releasing the touch releases the particles and the sound cluster dissolves.

Magnetic mode (no. 16 and Fig. 2(b)) induces attractive forces around each touch point. Particles that get into the attraction field do not perform their Brownian motion any longer but head directly toward the touch point. Releasing the touch relieves the particles that switch back to standard Brownian motion. The *repel* mode (no. 17 and Fig. 2(c)) is the opposite of magnetic mode. Particles within a certain radius around the touch point head straight away from it until they get beyond that radius or the touch is released.

Accentuate mode (no. 15 and Fig. 2(d)) is the only mode that does not affect the particles' motion but their loudness. Two volume levels can be set by sliders no. 9 and 11: accentuation level and standard level. Accentuate mode causes those particles under a touch point and within its effect radius to set their loudness to accentuation level.

In each mode, touches affect a certain field around the touch point. The size of this field is set by an interface element we call *bucket*. Buckets are activated like standard buttons. In addition to this, buckets can be "filled" by enlarging the inner circle up to outer circle size via pinch or drag gestures. The inner circle represents the size of the field around the touch. Thereby, different radii can be defined for the different interaction modes which is especially important for a further feature, mode combination.

It is possible to have several modes active at a time. The only two impossible combinations are repel mode together with magnetic mode or drag mode. All other combinations are possible and allow a number of interesting interactions with the particles and the stereo noise field they create.

In addition, all interaction modes can be assigned to specific frequency bands, i.e., to whole rows of the playground (no. 19). Once activated, a frequency band emits attractive or repelling forces, binds particles, or accentuates those crossing the band. The magnetic and repel radii set in the buckets apply to the band effects, too. Accentuation and dragging affect only those particles within the band.

3. PLAYING THE INSTRUMENT

This Section gives an overview of the sonic artifacts that the instrument produces. Its basic material is noise clusters created via additive synthesis of a number of sine signals. Few signals, spread over a wide frequency range, are still discriminable tonal instances. But with a growing density, the sound fades into an unsteady noise spectrum. Its agility depends on the speed and angular behavior of the particles. Very low settings create a relatively steady, slowly evolving sound. High values result in brisk and turbulent sounds.

Agglomeration areas establish sound clusters that become increasingly tonal the smaller the areas are. This effect can be achieved by magnetic and drag interaction, by the corresponding frequency band functions and by setting the playground borders very tight. Drag and magnetic mode further enable shifting of particle clusters. After releasing the particles, the cluster dissolves gradually in accordance to the Brownian motion characteristics. This can be sped up by repel mode. More than this, repel interaction creates sonic holes the longer it forces the particles to escape from the zone around the touch point.

A combination of modes that prove convenient is drag+ magnetic mode. While magnetic mode attracts all particles within the corresponding radius, those particles that get into the drag radius do not just follow the touch but can be dragged (which is usually faster than the standard particle speed). This way a big number of particles can easily be collected and moved. If additionally the accentuation mode is activated with a radius similar to drag radius and an accentuation level set to zero, the cluster disappears and becomes audible only after releasing the touch. This is a typical example where the different radii for each interaction mode plus the combination of several modes allows for a very nuanced complex interaction. However, the radius setup via buckets is not optimal as it takes relatively long.

Drag interaction comes with a further interesting sonic feature. Those particles being dragged faster than the standard speed become conspicuous even though their volume level is the same as that of the other particles. This effect intensifies the more particles are dragged. This can be a convenient tool for creating quasi-melodic phenomena.

Another means for emphasizing particles is accentuation. This interaction does not affect the particle motion but their volume level. It can be used in both ways, raising and lowering the volume level of the affected particles, respectively. This allows for the creation and modulation of spectral centroids. The accentuation level, just like the standard volume level, can also be set to zero to create holes in the spectrum. Applying accentuation to whole frequency bands can be used to define chord-like sound structures.

4. DISCUSSION AND FUTURE WORK

In our tests, the ideal hardware for running *TouchNoise* turned out to be tabletop-sized displays of about 32". Our development device features a FullHD resolution which is very convenient. The interface also scales down for lower resolutions and runs on smaller displays. We tested it on tablet-sized devices. But interaction becomes less nuanced due to reduced space for touch gestures.

Visually, the particle playground is well suited for projection during live performances. More than this, and as a remark from our personal experiences during testing, it invites very much to experiment. Beyond the basic effects described in the previous Section, there is much room for combinations and different configurations, all creating distinct sonic results. However, planning a real composition or live performance requires getting used to the particles' behavior. This includes, for instance, the time they need to spread over the playground, fill a sonic hole after releasing a repelling touch or get collected by a magnetic touch. The basically random motion of the particles complicates scripting and exact reproduction of compositions, at least as far as they are given as sequences of all too precisely defined gestures, e.g., "collect the upper right two particles and drag them down".

TouchNoise is no instrument to trigger musical events directly but an interface to affect the behavior of sonic particles. However, some interaction modes also for faster, very direct sound manipulation, e.g., drag and accentuate mode. Therefore, it could be useful to assign different interaction modes to different touches, e.g., to have some touches creating particles and others attracting and removing them again. This might be implemented via marker-equipped tangibles on a tabletop or via user-tracking on a wall where modes are assigned to each user.

One particular problem that we experience is that the particles immediately switch back to their standard behavior after releasing a touch. It is hard to create longer lasting structures. We plan to add vector field and swarming functionalities to achieve this in the next development stage.

We demonstrated *TouchNoise* to several musicians and composers already during the development and collected their suggestions. Most of them are already implemented or discussed previously. A question that came up regularly asked for some kind of tonality or musical scales. Following this, we plan to experiment with pitch quantization to the particles' output. We further consider the output of *TouchNoise* not necessarily as the end of the synthesis pipeline. By adding a keyboard-tracked comb filter, the spectral behavior and qualities of the system can also be transfered to more tonal sounds.

5. SUMMARY

Noise plays an important role in electronic music. Noise modulation so far is mainly based on filters or granular synthesis methods. With our digital musical instrument *TouchNoise* we investigate a different approach to noise modulation which works with autonomous sound particles, each generating a sine signal, added up to create the noise spectrum. Their distribution and movement in the two-dimensional frequency and stereo panning domain are based on Brownian motion and affected by multitouch interaction. Particles can be dragged, attracted, repelled, and accentuated by touches. The same effects can also be assigned to whole frequency bands. The particles' motion characteristics can be manipulated in many ways.

The feedback we gained during several demos and discussions is encouraging and inspires lots of further functionalities, as outlined in the discussion. These are subject to the next development phase. In parallel to this, we plan to invite composers to explore the instrument and its sonic possibilities which go beyond what we could outline in this paper.

6. REFERENCES

[1] T. Blackwell. Swarming and Music. In E. R. Miranda and J. A. Biles, editors, *Evolutionary Computer Music*, chapter 9, pages 194–217. Springer, London, UK, April 2007.

[2] J. Chadabe. Interactive Music Composition and Performance System. United States Patent Nr. 4,526,078, July 1985. filed Sept. 1982.

[3] R. H. Chapel. Realtime Algorithmic Music Systems From Fractals and Chaotic Functions: Towards an Active Musical Instrument. Master's thesis, University Pompeu Fabra, Department of Technology, Barcelona, Spain, Sept. 2003.

[4] M. Ciglar. An ultrasound based instruments generating audible and tactile sound. In *Proc. of New Interfaces for Musical Expression (NIME) 2010*, pages 19–22, Sydney, Australia, June 2010. University of Technology Sydney.

[5] A. C. Eldridge. Adaptive Systems Music: Musical Structures from Algorithmic Process. In C. Soddu, editor, *Proc. of the 6th Generative Art Conf.*, Milan, Italy, 2002.

[6] B. Eno and P. Chilvers. Bloom. app, March 2011.

[7] B. Eno and P. Chilvers. Scape. app, Sept. 2012.

[8] S. Gelineck, J. Andersen, and M. Büchert. Music Mixing Surface. In *Proc. of the 2013 Int. Conf. on Interactive Tabletops and Surfaces (ITS'13)*, pages 433–436, St. Andrews, Scotland, UK, Oct. 2013. ACM.

[9] S. Gelineck and N. Böttcher. 6to6Mappr: An educational tool for fast and easy mapping of input devices to musical parameters. In *Audio Mostly 2012: 6th Conf. on Interaction with Sound—Sound and Future Vision*, pages 117–123, Corfu, Greece, Sept. 2012. Ionian Academy, Interactive Institute/Sonic Studio Piteå, ACM SIGCHI.

[10] S. H. Hansen and A. R. Jensenius. The Drum Pants. In *Audio Mostly 2006: A Conf. on Sound in Games*, pages 60–63, Piteå, Sweden, Oct. 2006. Interactive Institute/Sonic Studio.

[11] T. Iwai. Electroplankton. Indies Zero, Nintendo, 2005.

[12] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces. In *Proc. of Tangible and Embedded Interaction*, Baton Rouge, Louisiana, USA, 2007.

[13] J. Rudess and T. Miller. SpaceWiz. app on iTunes, Aug. 2013.

[14] S. Sandler and J. Windl. NodeBeat. app for iOS, Blackberry, Android, and Windows, Jan. 2013.

[15] T. Unemi and D. Bisig. Music by Interaction among Two Flocking Species and Human. In *Third Iteration: Proc. of the Third Int. Conf. on Generative Systems in Electronic Arts*, pages 171–179, Melbourne, Australia, Nov. 2005. Monash University.

[16] B. Vera. Orbits—Generative Music. Android app, Oct. 2011.