TouchNoise: A New Multitouch Interface for Creative Work with Noise

Axel Berndt
Center of Music and Film Informatics,
University of Music Detmold,
Detmold, Germany
berndt@hfm-detmold.de

Nadia Al-Kassab
Interactive Media Lab,
Technische Universität Dresden,
Dresden, Germany
nadia.alkassab@gmail.com

Raimund Dachselt
Interactive Media Lab,
Technische Universität Dresden,
Dresden Germany
dachselt@acm.org

ABSTRACT

TouchNoise is a multitouch noise modulation interface designed for musical live performance. It allows the direct and indirect manipulation of sound particles in the stereophonic frequency spectrum. In order to increase TouchNoise's playability we conducted a comprehensive interface revision retaining only its core interaction concept. New interaction techniques and gestures for radial menus, effect range settings, and frequency band effects are introduced. The revision paved the way for a series of new functionalities, such as flocking, flow fields, and MIDI connectivity, making TouchNoise a fully-fledged, powerful interface for creative work with noise. This paper introduces the new TouchNoise interface and functionalities through a discussion of the revision process and derives interaction principles and design recommendations for musical multitouch interfaces in general.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies, Interaction styles*; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Signal synthesis*; J.5 [**Computer Applications**]: Arts and Humanities—*Performing arts*

General Terms

Design, Human Factors

1. MULTITOUCH, PARTICLES, NOISE

Multitouch interaction paved the way for making complex software systems more easily accessible. Multitouch gestures are often not only more intuitive than traditional keyboard and mouse input, but also faster and more direct. This, subsequently, can bring a gain of interaction flow and versatility, especially when bimanual interaction and multi-user scenarios are involved. In the field of musical human-computer interaction multitouch technology inspired many new concepts and interface approaches. Never before, complex and highly interactive sound synthesis methods, sequencing approaches and whole generative music systems were as easily accessible as nowadays.

With increasingly powerful computing technology a new type of digital musical instruments, the *active musical instrument*, emerged [5]. Such instruments do not require the player to trigger each musical event (e.g., note) individually. They play themselves autonomously in realtime. The player's role is to direct this process in a musically meaningful way. Among the numerous instances, we can name just a few examples here. A typical representative is the Reactable interface [9, 12], a multitouch and tangible interface for sound synthesis and sequencing. Its technical framework has been adopted in the tabletop algorithmic composition system ReacTacT [1]. CollideFx is a multitouch patching environment by Gnegy for realtime sound synthesis and effects processing [10]. Lopes et al.'s study with a multitouch DJing application attests an increase of interaction speed compared to a purely virtual (laptop) setup [15]. The NodeBeat app [22] is a dynamic sequencer on the cusp of an interactive ambient music generator like several smartphone apps by Eno and Chilvers [7, 19].

Our system TouchNoise [2] is a further example of a multitouch active musical instrument. It exploits new perspectives in the creation of and interaction with stereophonic noise spectra that were formerly impossible. It is based on an interactive particle system equipped with algorithms for different motion behaviors like Brownian motion, flocking and flow fields. Multitouch gestures not only set the parameters of these behaviors but also exert manifold direct influences to the particles, like magnetic and repellent forces.

Multi-agent and particle systems have been the basis of several musical interfaces. Kuhara & Kobayashi present a kinetic particle synthesizer for mobile multitouch devices [14]. Photophore is a synthesizer that applies a flocking algorithm to modulate up to 100 oscillators and create natural chorus effects [6]. Orbits is a generative music interface that is based on an intuitive simulation of the movement of orbs and gravitational forces between them [23]. A deeper discussion of the use of interactive swarming in an improvisational music context is delivered by Blackwell [3]. Artistic examples and C++ libraries for swarm-based music generation are provided by Bisig and colleagues.[1]

[1] `http://www.zhdk.ch/index.php?id=icst_swarms_e`, last access: July 2015.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AM15, October 07 - 09, 2015, Thessaloniki, Greece
Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3896-7/15/10 ...$15.00.
http://dx.doi.org/10.1145/2814895.2814896.

As a musical interface TouchNoise undergoes a development process in which we pursue the following general goals, based on [17]. The instrument's basic concept should be easy to understand, supported by a direct correlation of auditory and visual output, and basic interactions should be very direct with no practical learning hump. Mastery of the instrument's full functionality exploits increasingly manifold and complex sound patterns (open-endedness for long-term engagement, see [24]) and demands the player to develop advanced, more nuanced gestures and playing techniques (layered affordance). The visual impression should be suited for live projection on-stage and roughly promote comprehension and virtuosity to the audience.

In the first, exploratory development phase we implemented the basic mapping and interaction concepts [2], see section 2 for an overview. In practical tests and demo sessions we explored the sonic design space of the approach, gathered first usability experiences and collected initial user feedback. Based on this, we made substantial revisions (section 3) and extensions (section 4) during the second development phase. We believe that the experiences that we made during this process and the solutions that we came up with are applicable in similar contexts and may provide help and inspiration for recurrent design issues of digital musical instruments. Hence, we derive general design recommendations along the critical discussion (section 5). We see this work in a row with further works on the development of and design guidance for musical multitouch interfaces, such as Carrascal & Jordà's graphical user interface widgets for audio mixing [4].

This paper focuses on the user interface aspects of TouchNoise. It traces our design process and decisions, particularly from the second development phase. A deeper discussion of TouchNoise's sonic design space cannot be given in this paper and has to be the subject of a separate publication.

2. INTRODUCING TOUCHNOISE

For many centuries, the musical role of noise was mainly situated in the percussion section. Drummers and percussionists developed a great mastery in the rhythmical use of differently colored noise spectra and established various sophisticated playing techniques. The work with sustained noise sounds such as cymbal rolls, however, was less frequently. This changed significantly when the era of electronic music introduced manifold noise-based effect sounds. Wind is one famous example [11]. With the growing number of ever more flexible synthesis tools noise became one of the most important sound materials in electronic music nowadays.

Noise modulation is commonly based on stochastic signals synthesizing a somehow colored inharmonic frequency spectrum. Filters are then applied to further refine this spectrum. More complex frequency spectra, in fact any recorded sound, can be introduced via sampling and granular synthesis. All these approaches are well established and prove their practical worth over several decades of applications. When looking at how they are used in music making and how the corresponding synthesis tools are handled we recognize a prevalence of indirect interaction concepts. Sound manipulations are achieved by synthesis patch editing and modifications of frequency, amplitude and filter modulations plus certain distortion and waveshaping effects, all controlled via traditional control elements such as knobs, faders, and buttons. The mental distance between these control interfaces and the frequency spectrum that derives from them is relatively great.

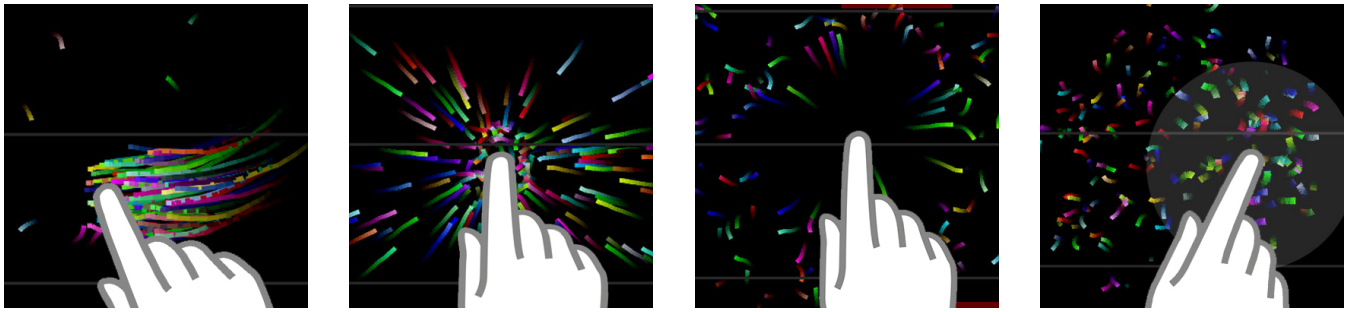
With TouchNoise we are looking for a more direct interaction technique to foster the creative and nuanced work with noise spectra and provide a live performable instrument that opens up new perspectives for this purpose. In this section, we introduce its basic concepts and describe its state at the end of the first development phase. TouchNoise is a multitouch interface, first introduced in [2]. It maps stereo panning and frequency domain onto the horizontal and vertical axis of a touchscreen and places sine oscillators, visually represented as points, on this 2d plane. We call them *particles*. The 2d plane is called the *particle playground*. If a particle moves along the horizontal axis, it changes its position in the stereo sound field. Movement along the vertical axis means the particle changing its frequency (between 20Hz and 20kHz). Usually, multiple particles are present on the playground. Their signals add up to the stereophonic frequency spectrum that the particle playground illustrates visually and that is immediately output. The particles are not static but by default perform Brownian motion [18] on the playground. The user/player can interactively add and remove particles to and from the playground. Most important, the user's multitouch gestures exert manifold influences on the particle motion, thus, on the distribution and dynamics of the noise spectrum.

Different functions can be assigned to touches, see figure 1. Thereby, it is possible to drag particles throughout the playground, attract and repel them to and from the touches, and accentuate their amplitude. Combinations of these functions are also possible, e.g., to attract particles and accentuate them when they come into a certain range. The radius of each effect can be set via an interface element that we call *bucket* (explained in figure 2). It is further possible to assign touch events with the creation or deletion of particles at and around the touch position. An upper and lower bar delimit the frequency axis of the particle playground and can be dragged to any position between 20Hz and 20kHz. These interactions influence the particle distribution and (amplitude) mixing and take place directly within the stereophonic frequency spectrum or its visual representation on the touchscreen, respectively.

There is, nonetheless, still a number of indirect interactions that aim at general characteristics of all particles. Two parameters control the Brownian motion.[2] The step width determines the particles' speed and the Brownian angle delimits the maximum rotation. These two parameters set the dynamic behavior of the noise spectrum (e.g., static, slowly evolving, or brisk). A lifetime slider sets the timespan that each particle exists on the playground before deleted automatically. A further slider allows the creation and deletion of particles at random positions.

The touch functions (drag, magnetic, repel, and accentuate) can also be assigned to whole frequency bands through a matrix of buttons (see figure 3). When activated, a frequency band exerts the same effects as the corresponding touch function, i.e., it attracts or repels particles above and below within a certain radius, accentuates particles crossing the band or holds them captive. The matrix can be regarded

[2]Brownian motion in 2d: rotate by a random angle, make a step forward, rotate by a random angle, make a step forward, and so on.



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

Figure 1: Modes of direct touch interaction with the sound particles.

as a "frequency band piano", thus, we will conveniently call it *piano* in the remainder of this paper. Its vertical resolution determines the number of frequency bands (keys on the piano, respectively) and their size as the piano always cover the full range from 20Hz to 20kHz. This results either in a few wide bands or in many narrow bands. The resolution and activation time of the keys can be adjusted by the user. Once activated, the frequency bands may either remain active until the user deactivates them or they deactivate automatically after the adjusted activation time.

All control elements are hidden in a left and a right panel to keep the screen free and reserved for the particle playground and the touch interaction within it. This reflects our intention to facilitate a direct interaction with the stereo noise spectrum. The panels, when required, can be dragged into the screen and dragged or flicked away afterward. A full screenshot of the graphical user interface at the development stage explained so far is shown in [2] and cannot be reprinted here due to lack of space.

The TouchNoise version described in this section has its worth mainly as a concept study. It demonstrates the manifold novel interactions with noise spectra. Its functionality is easy to comprehend. Sonic and visual output feature a direct and intuitive correlation. During several practical test and demo sessions with lay and expert users over the course of one year we discovered potential playing techniques, identified several usability issues and weaknesses, and made revisions and extensions. The succeeding sections describe how TouchNoise grew during this iterative process.

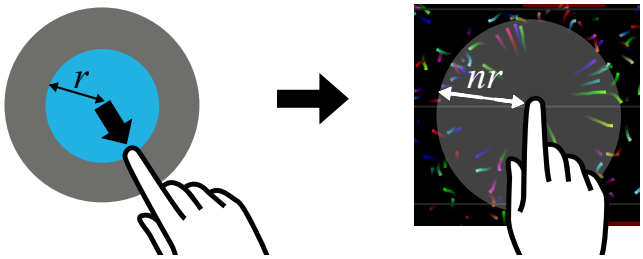


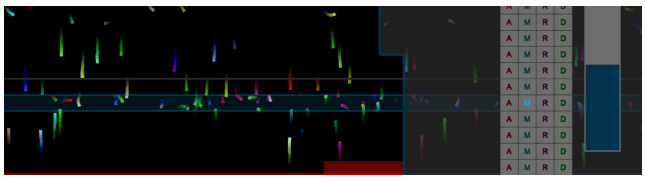
Figure 2: Buckets can be (de-)activated like buttons. Additionally, they allow setting the effect radius of touch interactions by enlarging the inner part via drag gestures. Because of the bucket's fixed size it is hard to foresee the actual effect radius on the playground.

3. EXPERIENCES AND REVISIONS

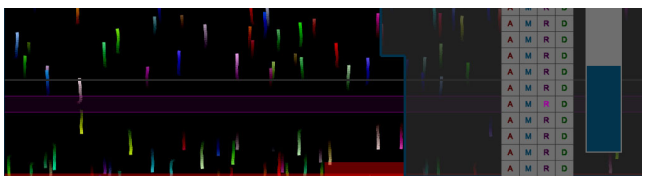
Our main design goal with TouchNoise was the direct interaction with the stereo noise spectrum. Thus, we reserved the full screen space for the particle playground and the gestural interactions on it (drag, attract, repel, accentuate). All other interface elements were hidden in two panels that can be dragged into the screen when required. The left panel contains the controls for the creation and deletion of particles and their motion characteristics, the right panel contains the buckets to activate the direct interaction modes, set their radii, and assign modes to frequency bands, i.e., the piano. It turned out during our practical test and demo sessions that both panels are used very frequently. When dragged into the screen each panel covers nearly half of the playground and hampers interaction on it. Hence, we experienced an undesired shift toward indirect interaction. During the revision phase, we put more emphasis on the direct interaction, i.e., eliminate the overfull panels, make the interface elements more directly accessible, reduce their complexity, and link them stronger with the playground, especially the sprawling, fiddly piano matrix.

The new interface design is shown in figure 4. An accompanying demo video shows the interface at work.[3] We

[3] `http://youtu.be/9z4Fer8blKA`



(a) Magnetic frequency band.



(b) Repellent frequency band, comparable with a notch filter.

Figure 3: Interaction modes (here magnetic and repel) can be assigned to frequency bands.

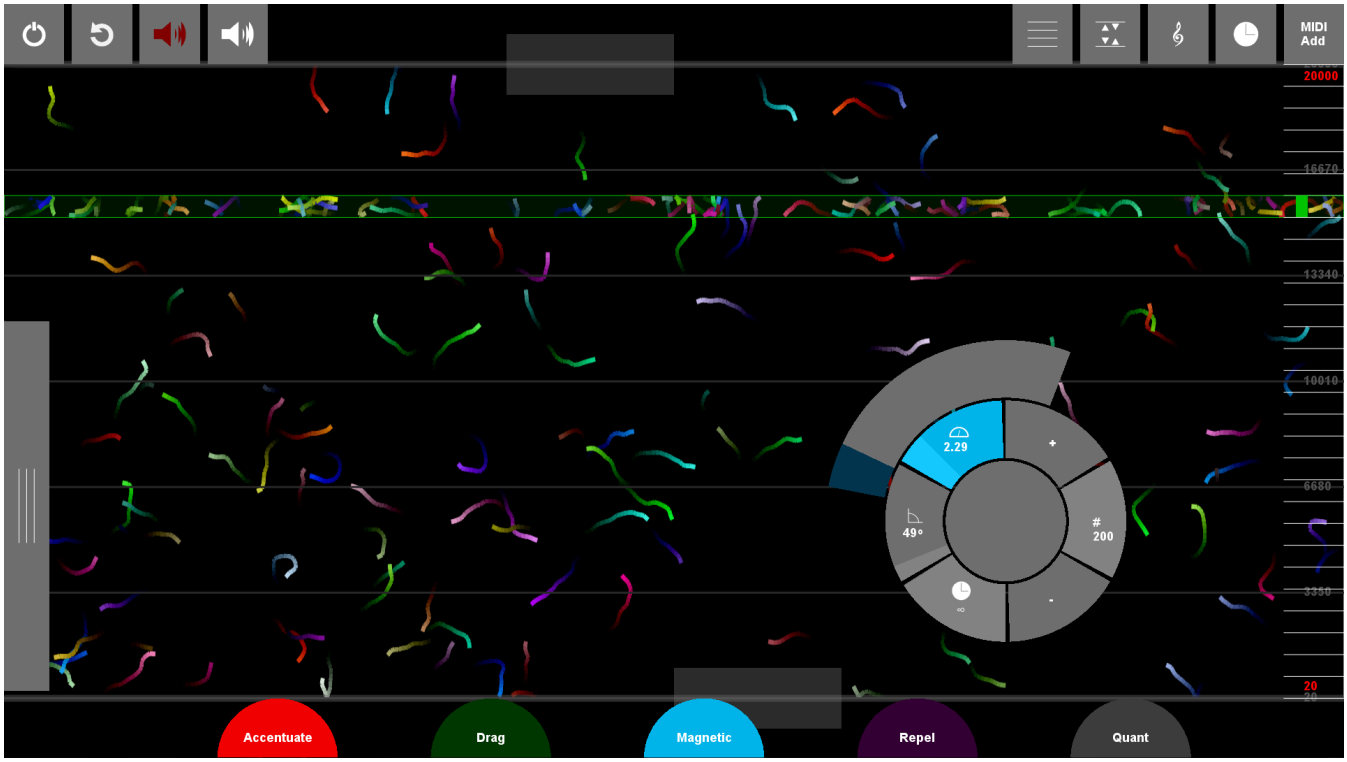


Figure 4: A screenshot of TouchNoise's new graphical user interface. A frequency band has an active drag function. Accentuation and magnetic mode are active for direct touch interaction within the particle playground. The circular particle menu is also open; it can be freely dragged over the screen and minimizes when put at or flicked to the border.

started with restructuring and grouping those elements that belong together based on related functionality. Frequently used elements are placed closer to the user, easy to reach and use. This applies particularly to the buckets and the particle controls which are now placed on the bottom edge of the display. Volume controls, piano bar settings, exit and reset button were placed on the upper display edge.

The buckets suffered from a scale conflict. Because of their fixed size it is difficult to predict the effect radius on the playground from the "fill level" of the bucket, cf. figure 2. Our redesign eliminates this conflict, see figure 5. Radius setup with our new *effect range and toggle widget* is done by a drag gesture starting within the button area of the widget (the semi circles at the bottom margin of figure 4) and then moving out of it. A transparent circular field indicates the effect range that the interaction will have on the playground

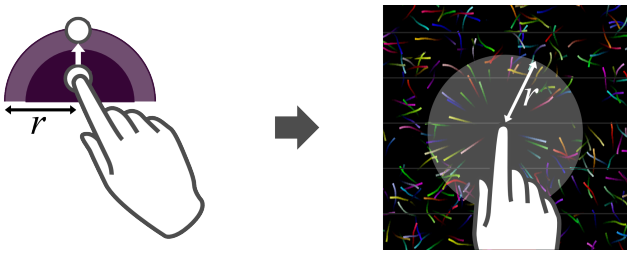


Figure 5: Radius setting with an *effect range and toggle widget*. The effect range that the interaction will later have on the playground is indicated at a 1:1 ratio.

at a ratio of 1:1. Releasing the touch sets the radius and the transparent field disappears. Activation and deactivation of the effect is done by a simple tap into the button area.

The particle settings are indirect by nature. They are grouped in the particle menu. It must be compact to allow fast, convenient settings and to be least interfering to the interaction on the playground. This led us to the design of a radial menu (see figure 4 in the right area) that can freely be placed on the screen and minimized by dragging or flicking it to a display edge. It opens when it is dragged back into the screen. Besides two buttons all elements in the particle menu are sliders which also cling to the circular design. The Brownian angle slider is a complete circle around the menu. The others are only partial circles, just as the particle speed slider that is opened in figure 4. The sliders are opened by a tap on the corresponding menu icon or by a continuous drag gesture that starts on the menu icon and, without releasing the touch, directly moves to the slider value to be set. A similar continuous gesture for radial sliders has been described by Kister et al. [13]. This convenience feature speeds up the work in this menu and makes interaction in this place more efficient. The radial slider design comes along with another advantage. When the drag gesture to set a certain slider value moves further out of the circle it becomes easier to make very fine adjustments as the arc length increases for the same angular motion. It is also possible to change the menu's size via a pinch gesture in the center of the particle menu.

The piano matrix went through an extensive redesign, too. The new piano bar is placed at the right display border.

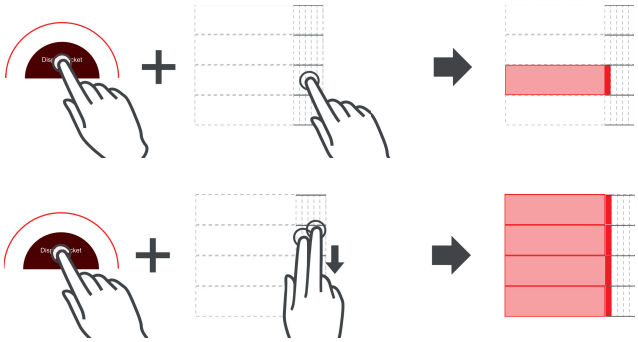


Figure 6: Bimanual gestures to assign effects to frequency bands.

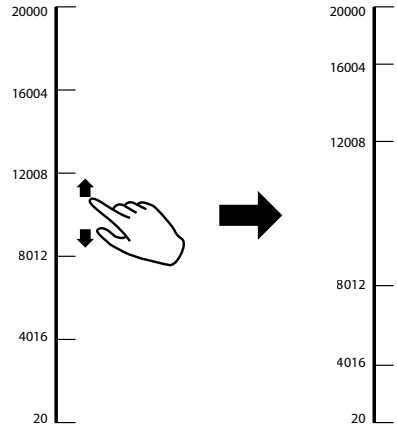


Figure 7: Frequency mapping distortion via pinch gesture.

Its visual design is minimalistic and unobtrusive, indicating only the frequency bands. Thus, there is no need anymore to hide it in a panel. The sprawling matrix of buttons, necessary to activate any effects on the bands, turned out to be redundant since the effects are already represented by the effect range and toggle widgets. Assigning one of the touch effects to a frequency band is now done by a bimanual gesture: tap and hold one or more effect range and toggle widgets and tap or drag over the piano bar to assign or deassign the effect. The assignments are then visually indicated by color coding (same color as the widget), as shown in figure 6. If the drag gesture over the piano bar is done with two fingers, the effect is assigned (drag down) or deassigned (drag up) to all bands. The new piano bar allows for a more fluid interaction than the fiddly piano matrix did. It is more directly linked to the particle playground without covering any screen space or hampering interaction on the right side of the playground.

Another problem turned out to be the lack of possibilities to define longterm behaviors. Once a touch is released the particles switch back into standard Brownian motion mode. The particle distribution fades gradually into white noise as far as the piano exerts no further influence. This can be significantly decelerated by a slow particle speed and great Brownian angle settings. However, as an active musical instrument TouchNoise should not just fade back into a basic sound but allow for more directed and more complex longterm behaviors with regard to the particle flow and distribution.

Frequent requests also asked for the introduction of musical scales, i.e., quantization over the frequency domain so that the basic tonality of TouchNoise could also be based on musical pitches instead of noise or even a mixture of both. Functions for longterm behavior and quantization are described in the succeeding section together with a series of further new functionalities.

4. EXTENSIONS

Several extensions of TouchNoise follow from our practical experiences and the demo feedback, as described in the previous section. These aim for both a more nuanced definition of the frequency domain and the creation of persistent particle flow characteristics.

4.1 Frequency Domain Extensions

The initial mapping of the vertical axis of the display, resp. playground, to the frequency domain is linear. Assuming a uniform distribution of the particles over the playground, the linear mapping causes an emphasis of higher pitches since the human subjective perception of pitch is approximately logarithmic in the frequency domain [8, 16]. Thus, the noise sounds denser in higher pitches. A musical use of TouchNoise and especially of the piano suggests a logarithmic frequency mapping which is now accessible via a toggle button. When switching from one mapping to the other, the particles are replaced at the new vertical position of their current frequency to avoid sonic inconsistencies through frequency jumps.

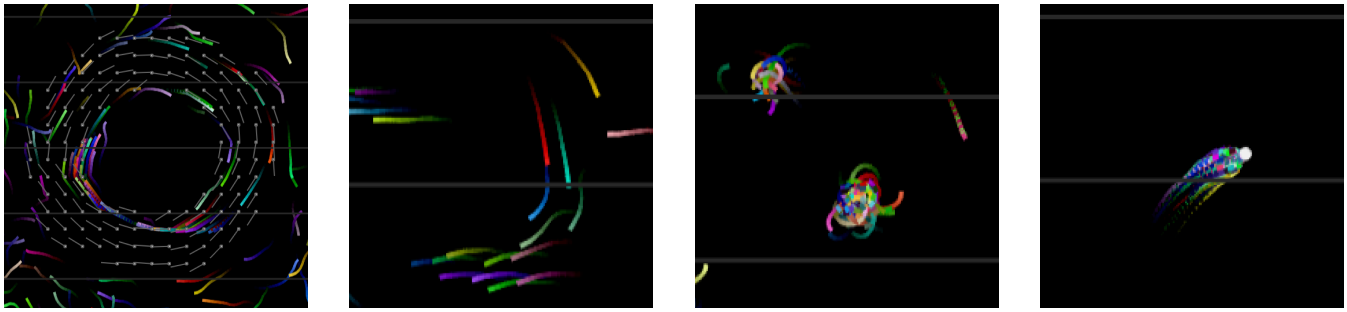
Further nonlinear distortions of the frequency domain are possible. The user can expand frequency bands by pinch gestures, similar to a fisheye zoom (see figure 7). While the particles still move visually with the same step width, the actual frequency steps vary according to the mapping of the visual position to the frequency position. The particles pass visually narrow frequency bands much faster than those being widened by pinch gestures or logarithmic mapping. This affects the distribution of the particles in the frequency domain.

We introduced quantization as an additional effect that can be assigned to frequency bands. Each particle that enters such a band plays its median frequency until it leaves the band again. In addition to the possibility of adjusting the number of frequency bands, the user can also choose between predefined musical scales: pentatonic scale, Pythagorean diatonic scale, and the equal tempered chromatic scale.

We further added a MIDI interface that makes it possible to play the frequency bands via a controller keyboard. Any combination of effects can be assigned to MIDI via the same bimanual gesture as for the piano bar. The MIDI button is placed at the top right of the screen above the piano bar. It toggles between three modes.

Key mode assigns the frequency bands band-wise to the keys of the MIDI keyboard. This is most intuitive for the chromatic scale. Keystroke activates the frequency band, key release deactivates it.

Freq mode computes the frequency of the MIDI notes. A keystroke activates the frequency band that includes this frequency. Activation and deactivation is then similar to the key mode.



(a) A vector field can be used to direct the particle motion.

(b) Flocking by alignment.

(c) A strong cohesion causes the particles to stick together.

(d) The white leader particle attracts particles nearby.

Figure 8: Modes of particle flow.

Add mode is not connected to the frequency bands. It creates some particles at the pitch position of the struck MIDI key and deletes these particles when the key is released.

Thereby, it is possible to play a MIDI keyboard and perform interactions within TouchNoise at the same time. This makes the work with frequency bands faster and more versatile. A typical scenario could, e.g., be the following. The main volume is muted, particles become audible only when accentuated. Accentuation is assigned to the MIDI input in key mode and the chromatic scale is set. This transforms the MIDI keyboard into a series of bandpass filters, one bandpass per key, that are activated by keystroke. It is now possible to play musical notes on the spectrum that the particles on the playground define. Multitouch gestures on the playground exert further influence to this spectrum, i.e., the distribution and motion of the particles in the frequency and stereo field.

4.2 Particle Flow Extensions

The extensions related to the particle flow address the issue of missing longterm behavior in two regards: the path of the particle movement and the particle distribution or clustering.

Up to now, the particle flow is defined by the parameters of the Brownian motion and influenced by magnetic, repellent and drag interaction through touches or frequency bands. The particles switch back to Brownian motion when the interaction ends. This, however, does not suffice to define longer lasting directed flow which is sometimes desired, as discussed in section 3. Therefore, we introduce vector field functionality to the playground. The vector field can be defined, literally painted, by drag gestures on the playground. Figure 8a shows an exemplary circular field. A particle that gets into it follows the indicated directions until it is released to the omnidirectional area where it switches back to Brownian motion.

We added different flocking functionalities to introduce persistence also to the clustering of particles. It is possible to add a leader particle to the playground that attracts local particles, see figure 8d. Moreover, flocking and swarming are even possible without a leader through the Boids algorithm by Reynolds [20, 21]. It gives us control over the three parameters separation (avoid crowding local particles), alignment (head toward the average direction of local particles, see figure 8b), and cohesion (head toward the average position of local particles, see figure 8c).

All these influences—vector field, leadership, and Boids—can be weighted against each other and the Brownian motion. This allows for the creation of complex behaviors, e.g., a flock that follows a leader but the flockmates keep a certain minimal distance to each other and exhibit a certain degree of Brownian behavior even within the flock. The whole flock may follow a path through a vector field, but only roughly because of a very low vector field weight. Touch interaction, however, has highest priority and dominates all other motion influences.

For the tuning of these weightings we added a further element to the graphical user interface, the *flip back panel*. It is shown in figure 9. In figure 4 it is hidden, only its handle is visible at the left border. The sliders in this panel indicate the weightings. Their settings are typically done once in a wile and then interaction returns to the playground. Following this, we decided to eliminate the necessity of closing the panel, i.e., dragging it back off the screen. Instead, the user opens it with the drag gesture that is typical for panels and holds it while adjusting the sliders. It then flips back automatically when released. The layout of the linear sliders is conceived so that several sliders can be manipulated at once with multiple fingers of one hand.

5. DISCUSSION AND FUTURE WORK

In our practical test sessions with several lay and expert users we experience a more fluid and versatile interaction with the new TouchNoise interface. This is, however, only a subjective perception that still needs to be objectified in more extensive user tests. The main reason for our experience so far is most likely the faster, more direct accessibility of all interface elements which do not overlay the playground as broad as the previous panels and foster an immediate return to the playground (e.g., the flip back panel). More interaction takes place on the playground reinforced by the new functionalities, especially the vector fields, but also by several new interaction gestures. While setting the radius of an effect range and toggle widget, the hand moves already over the playground and is there right when the setting is done and the direct interaction with the particles starts. Also the new piano bar at the right edge of the playground and the dedicated bimanual gestures contribute to this im-

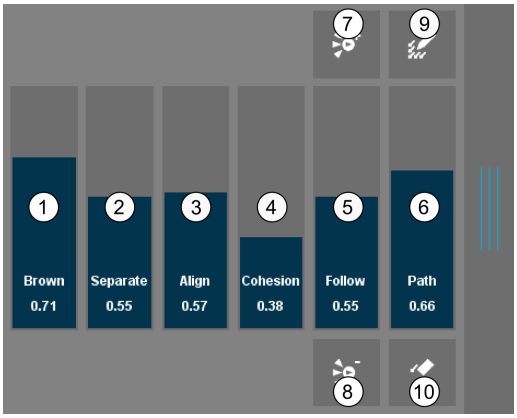


Figure 9: The weightings of the particle behaviors (1–6) are adjusted in this flip back panel. It has to be held open and flips to the left screen border when released. Buttons 7 and 8 toggle the creation and deletion of leader particles, buttons 9 and 10 the drawing and erasing of the flow field arrows.

provement. No panel has to be opened to play the piano bar. The MIDI extension provides an even more versatile work with it, though it is off the screen.

We, nonetheless, register several further issues and limitations that are interesting to discuss at this point.

The maximum number of particles on the playground is constrained by hardware performance. Our development system (Intel Core i7-3770 CPU (3.40GHz), 16GB RAM, a GeForce GTX 680 graphics card, a standard sound chip, and a 3M multitouch screen) allows up to 200 simultaneous particles, glitch free, including their simulation, graphical rendering, sound synthesis, and interaction. This is just enough for a multifarious work and a rich sonic design space. We experimented with up to 400 particles which brings a gain of quality to some sound effects due to the stronger diffusion that more particles can achieve. With improving future hardware this limitation can easily be adapted.

A layout-related issue is the distance between the effect range and toggle widgets. A strong point about TouchNoise is its openness to manifold combinations of effects to create complex sonic behaviors. Hence, the user de-/activates multiple widgets often at once, for instance, to quickly assign them to frequency bands. Concerning our experiences on 27 inch and 32 inch touchscreens, which marks our suggested optimal screensizes, the effect range and toggle widgets should be placed closer to each other so that any combination of them is in reach of a single hand.

Another limitation arises when TouchNoise is played rhythmically. It is possible to perform rhythmic patterns, e.g., with accentuation touches. Currently, all envelope transitions are relatively smooth and inappropriate for rhythmic accents with very short and energetic attacks. This long attack phase may also be perceived as latency in addition to that of the touch and gesture recognition. Both together hamper a nuanced rhythmic playing. Chimes-like percussion effects can be achieved by continuously creating particles with a lifetime of only 100 milliseconds. The piano, too, allows only a reasonable rhythmic playing. Hence, a future work is to experiment with shorter attacks or envelope transitions, respectively, and identify the interaction contexts in which this is more desirable than our current, smooth transitions. Our expectation is that this is the case for accentuations and piano interactions but not for the creation and deletion of particles and for volume settings in general.

In many situations less than the maximum number of particles is audible. Only a few particles may have been created. The main volume may be zero and only accentuated particles are output. In these situations the loudness level is significantly lower than a full tutti, even with a maximized accentuation level. Practical tests have to show whether this is desired or if a subsequent dynamic range compression should compensate this to a certain extent.

The strengths and weaknesses discussed in section 3 and in this section, which were the basis of our revisions and extensions, are based on our own practical experiences and third-party feedback from several test and demo sessions over the course of one year. From these experiences and our consequential solutions we can derive some general interaction principles and design recommendations for multitouch-operated digital musical instruments. Musical instrument playing is a very direct and immediate interaction. The same should apply to digital musical instruments. Multitouch technology complies with this requirement. A user interface design that features a direct and immediate correspondence of visual and auditory feedback, such as the visual representation of the stereo sound spectrum in TouchNoise, facilitates an intuitive interaction with low learning humps. Frequently used interface elements should be directly accessible. Even the nesting within a directly accessible panel was a noticeable barrier—menus are even worse. Mutual coverage of interface widgets hampers their direct accessibility, interrupts interaction flow and should be avoided. If not avoidable, make the widget movable (particle menu) or automatically vanishing when interaction with it is finished (flip back menu). Often used elements should be placed closer to the user and easily, at best blindly, accessible. Therefore, the layout should exploit human spatial recognition capabilities. Continuous gestures facilitate interaction flow which is particularly beneficial for musical interaction. Complex functionality can still be rapidly accessible via bimanual gestures. These can reduce the necessity for additional graphical interface elements dramatically (piano bar instead of piano matrix) and fosters a minimalist design. From the bucket widgets we learned that a 1:n ratio for the effect range settings was unintuitive and completely impracticable. With the *effect range and toggle widget* we introduce a widget that features a 1:1 ratio.

Our next steps will focus on field experiences. We plan to invite musicians and composers to get in touch with TouchNoise, explore its manifold possibilities, develop playing techniques, and conduct sound and music aesthetic experiments. From this we hope to gain more objective feedback and clues for the next development phase.

6. CONCLUSIONS

In this paper we reflected on the development of and interaction with TouchNoise, a multitouch interface for noise modulation, an active musical instrument that is based on a multi-agent system of sine oscillators within the stereophonic frequency spectrum. Not only does it constitute a new way of interacting with noise spectra with manifold new modulation possibilities. With TouchNoise we also developed a visual mapping that enables direct interaction with the sound.

After the first development phase we evaluated our practical experiences from test and demo sessions and improved the interface substantially with extensive revisions and extensions. This reinforces the direct interaction with the stereo spectrum, the immediate accessibility of all functionalities, and the interaction flow. The direct interaction with a digital musical instrument and its sound or timbre, respectively, is a recurrent matter in many related projects. Against this background the interaction principles and design recommendations that we derived from our work on TouchNoise may also help and inspire similar projects on multitouch-operated musical interfaces.

We had to exclude a deeper discussion of the sonic design space that TouchNoise spans as this would have led beyond the constraints of this paper. Some first impressions are given in [2] and in the accompanying demo video of this paper. Further systematic discussion on the sounds of TouchNoise and playing techniques will be subject to future work and publications.

7. REFERENCES

[1] C. Ó Nuanáin and L. O'Sullivan. Real-time Algorithmic Composition with a Tabletop Musical Interface—A First Prototype and Performance. In *Audio Mostly 2014: 9th Conf. on Interaction with Sound—Imagining Sound and Music*, Aalborg, Denmark, Oct. 2014. Aalborg University, Interactive Institute/Sonic Studio Piteå, ACM.

[2] A. Berndt, N. Al-Kassab, and R. Dachselt. TouchNoise: A Particle-based Multitouch Noise Modulation Interface. In *Proc. of New Interfaces for Musical Expression (NIME) 2014*, pages 323–326, London, UK, June 2014. Goldsmiths, University of London.

[3] 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.

[4] J. P. Carrascal and S. Jordà. Multitouch Interface for Audio Mixing. In *Proc. of New Interfaces for Musical Expression (NIME) 2011*, pages 100–103, Oslo, Norway, May 2011. University of Oslo, Norwegian Academy of Music.

[5] 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.

[6] N. Dika. Photophore Synth. Taika Systems Ltd., Dec. 2014. app on iTunes, v1.0.

[7] B. Eno and P. Chilvers. Scape. Opal Limited, June 2014. app on iTunes, v1.1.

[8] H. Fastl and E. Zwicker. *Psychoacoustics: Facts and Models*, volume 22 of *Information Sciences*. Springer, Berlin, Heidelberg, Germany, 3rd edition, 2007.

[9] G. Geiger, N. Alber, S. Jordà, and M. Alonso. The Reactable: A Collaborative Musical Instrument for Playing and Understanding Music. *Her&Mus. Heritage & Museography*, pages 36–43, 2010.

[10] C. N. Gnegy. CollideFx: A Physics-Based Audio Effects Processor. In *Proc. of New Interfaces for Musical Expression (NIME) 2014*, pages 427–430, London, UK, June 2014. Goldsmiths, University of London.

[11] J. M. Jarre. Oxygène. Disques Dreyfus, Polydor, Dec. 1976. music publication.

[12] M. Kaltenbrunner, S. Jordà, G. Geiger, and M. Alonso. The reacTable: A Collaborative Musical Instrument. In *Proc. of the Workshop on "Tangible Interaction in Collaborative Environments" (TICE), at the 15th Int. IEEE Workshops on Enabling Technologies*, Manchester, U.K., 2006.

[13] U. Kister, P. Reipschläger, and R. Dachselt. Multi-Touch Manipulation of Magic Lenses for Information Visualization. In *Proc. of the Int. Conf. on Interactive Tabletops and Surfaces (ITS) 2014*, pages 431–434, Dresden, Germany, Nov. 2014. Interactive Media Lab Dresden, Technische Universität Dresden, ACM.

[14] Y. Kuhara and D. Kobayashi. Kinetic Particles Synthesizer Using Multi-Touch Screen Interface of Mobile Devices. In *Proc. of New Interfaces for Musical Expression (NIME) 2011*, pages 136–137, Oslo, Norway, May 2011. University of Oslo, Norwegian Academy of Music.

[15] P. Lopes, A. Ferreira, and J. A. M. Pereira. Battle of the DJs: an HCI perspective of Traditional, Virtual, Hybrid and Multitouch DJing. In *Proc. of New Interfaces for Musical Expression (NIME) 2011*, pages 367–372, Oslo, Norway, May 2011. University of Oslo, Norwegian Academy of Music.

[16] G. Loy. *Musimathics: The Mathematical Foundations of Music*, volume 1. MIT Press, Cambridge, Massachusets, June 2006.

[17] J. McDermott, T. Gifford, A. Bouwer, and M. Wagy. Should Music Interaction Be Easy? In S. Holland, K. Wilkie, P. Mulholland, and A. Seago, editors, *Music and Human-Computer Interaction*, chapter 2, pages 29–47. Springer, London, UK, Mar. 2013.

[18] E. Nelson. *Dynamical Theories of Brownian Motion*. Princeton University Press, Princeton, NJ, USA, 2nd edition, Aug. 2001.

[19] S. O'Neill and P. Chilvers. Air. Opal Limited, Feb. 2012. app on iTunes, v1.3.

[20] C. W. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, July 1987.

[21] C. W. Reynolds. Boids: Background and Update, July 2007. accessed: 20 Jan. 2015.

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

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

[24] I. Wallis, T. Ingalls, E. Campana, and C. Vuong. Amateur Musicians, Long-Term Engagement, and HCI. In S. Holland, K. Wilkie, P. Mulholland, and A. Seago, editors, *Music and Human-Computer Interaction*, chapter 3, pages 49–66. Springer, London, UK, Mar. 2013.