Kjetil Falkenberg Hansen
and Roberto Bresin
Kungliga Tekniska Högskolan (KTH
Royal Institute of Technology)
Computer Science and Communication
Lindstedsvägen 24, 10044 Stockholm,
Sweden
{kjetil, roberto}@kth.se

The Skipproof Virtual Turntable for High-Level Control of Scratching

Skipproof is an application that emulates a typical disc jockey (DJ) setup of turntable plus mixer and also allows high-level control of the playing style known as scratching. High-level control in this case means performing with modeled, complex DJ gestures through simplified actions: For instance, letting a single movement produce a sound that normally would require precisely synchronized right- and left-hand gestures. The performer controls Skipproof either with the software interface or through hardware devices connected to the computer. The hardware devices become alternative performance interfaces to the standard turntable, controlling the Skipproof application with both low-level gestures and high-level control actions in real time. The mapping between hardware input and Skipproof output is freely adaptable. Skipproof is in the prototype phase, but it has already been used in several projects in recent years.

Traditionally, scratching is performed through synchronized gestures: One hand controls the record speed on the turntable (thus also the pitch), and the other hand uses the audio mixer's crossfader to turn the sound on or off. The crossfader is a slider that was originally designed for fading gradually between two turntables (or other sound sources) in order to go seamlessly from one song to the next, but in scratching it is instead used for turning the sound of a single turntable rapidly on or off (often while the other turntable is playing for instance a rhythm track). The playing gestures are commonly known as *scratch techniques* and constitute a common language among DJs (Hansen 2002; Smith 2006). Similarly to other instruments, traditional DJ playing skills must be acquired through dedicated practice, which is reflected in a growing market for teaching material (e.g., DJ Q-bert 2003; Sloly and Frederikse 2004; Webber 2007).

Earlier studies (Hansen 2002; Hansen and Bresin 2003) have analyzed and described DJ-performed scratch techniques, and some of the most popular techniques have been modeled and implemented in Skipproof. The modeling was based on this analysis, with additional input gained by having an active dialog with the scratch DJ community. These models can in turn be used and manipulated in real-time performance with simple control actions; this makes it possible even for non-experts to play expressively within the stylistic boundaries of DJ playing practices.

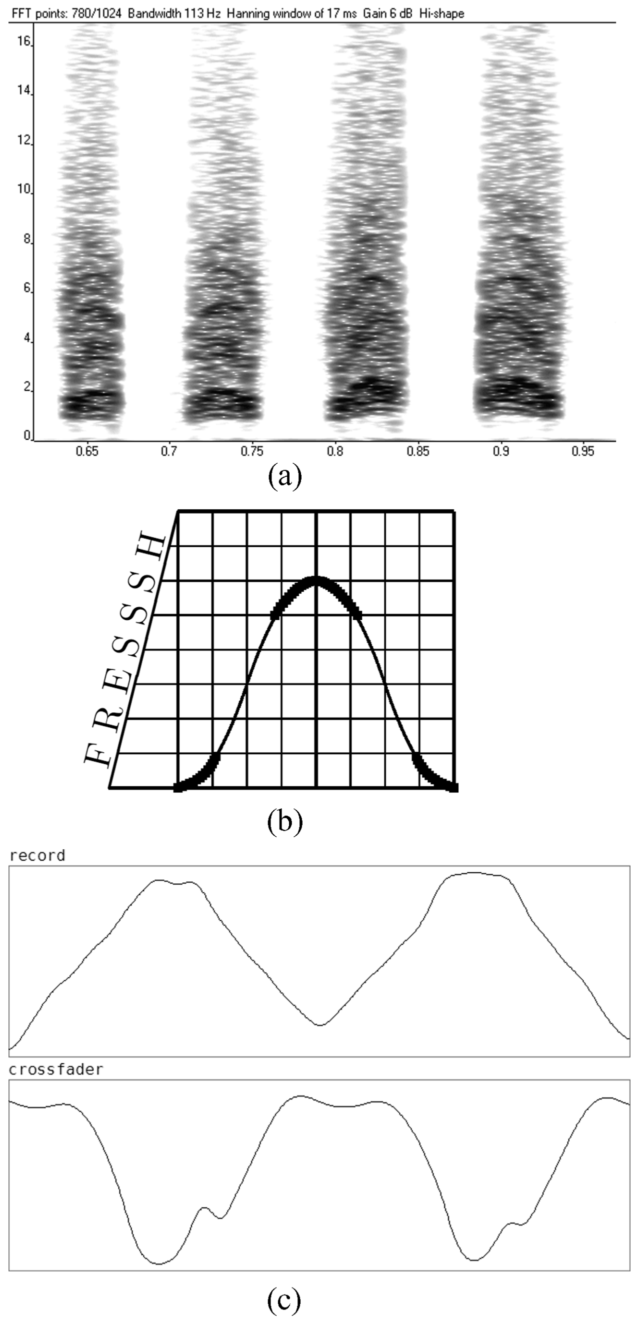
The motivations for writing Skipproof were to have a platform on which to model and simulate scratch techniques, as well as a tool for studying how scratch techniques are used in expressive performances. Also, we wanted to explore instrument-mapping strategies for scratching and to experiment with alternative performance interfaces for DJs. Skipproof can be combined with hardware and other software, and has been featured in performances with DJs using quite different control interfaces (including the Radio Baton, the Reactable, and various gesture sensors).

To the authors' knowledge, this is the first software that implements modeled scratch techniques and provides high-level performance control of such techniques. The paper is structured as follows: The following section gives a background on scratching and DJ interfaces, and then the Skipproof application is described. Three performance situations where Skipproof has been used are presented, including results from informal user evaluation. Finally, the current implementations and possible future uses of Skipproof are discussed.

Skipproof is available under the terms of the GNU Public License (GPL) and can be downloaded from http://www.speech.kth.se/~kjetil/software/.

Computer Music Journal, 34:2, pp. 39–50, Summer 2010

Figure 1. Three representations of the chirp scratch. (a) Spectrogram of two chirps in succession performed on the fresh sample. (b) Common graphical notation (Carluccio et al. 2000) of a chirp's record and crossfader movements: The plot shows the phonograph needle position relative to the fresh sample (vertical axis) as a function of time; the bold part of the movement is muted by the crossfader. (c) Sensor output from the record and crossfader of a recording used in Skipproof.



Background

A recent study of DJ performances describes the playing strategies and acoustics of the instrument (Hansen, Fabiani, and Bresin submitted). The study identifies how performance parameters are used expressively in scratching, and also some of the instrument's functional ranges. These and previous findings have been used to set parameters in Skipproof.

DJs move the record to manipulate the playback speed or pitch of the sound sample, but also to vary the playing position in the sample and change between samples. The crossfader is used to mute the sound temporarily, either to control tone durations, or to create short sound bursts or silent gaps down to around 10 msec. Onset properties, or tone attacks, are shaped both with the record and the crossfader to make, for instance, smooth attacks of otherwise sharp sounds. Generally for scratching, the crossfader is adjusted so that moving the fader only 1 mm will change the output from muted to maximum sound level. The fader knob can then easily be wiggled over the small area to give rapid onsets and offsets, generating a high tone density.

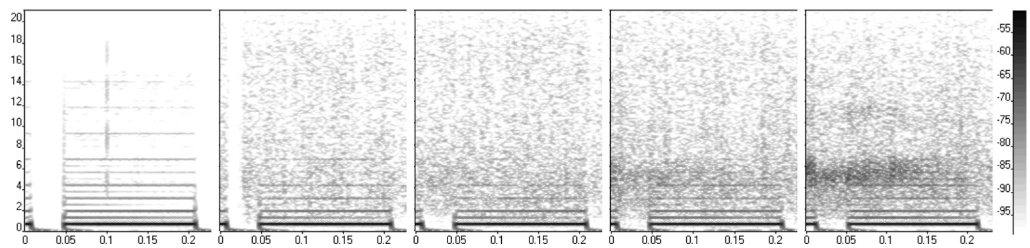
More than 100 scratch techniques have been described (DJ Q-bert 2007), and new ones appear regularly. The scratches are often given descriptive names, such as the *chirp* scratch: Chirps are performed in a series of rapid repetitions, where the record is moved in fast forward–backward motions and the crossfader is switched off and on to mute the sound at the start, at the turn, and at the end; see Figure 1b. The onset and offset of each sound are sharp, and the pitch quite stable and high because of the fast vinyl movements (Figure 1a), producing sounds that resemble chirping birds.

Traditional DJ Equipment

Record players were not originally intended to be used as musical instruments. Therefore, they have some less advantageous attributes, such as the vulnerability of the needle and pick-up system, the vinyl as sound storage, and, not least, the size and weight. A very basic setup of two turntables and a mixer weighs around 35 kg, and for a 5-hour set the vinyl albums needed could be approximately 50 kg.

Turntables are fragile in the sense that the needle can easily skip and even break. To avoid needle jumps and noise, the record should be undamaged,

Figure 2. Vinyl deterioration is visible in the five spectrograms when comparing a record in new condition (leftmost spectrogram) to the state after 1, 2, 12, and 15 minutes of scratching. For this particular vinyl, the level of wear was quite constant between approximately 2 and 12 minutes. From Hansen and Bresin (2003).



the pick-up arm correctly adjusted, and the table and floor free from vibrations.

A vinyl record has both disadvantages and advantages as a storage medium: For example, each side holds only around 10–20 minutes of music, but offers a good visual representation of the content and track positions by the center label, stickers, and groove density. The physical damage the needle inflicts on the vinyl is considerable (see Figure 2), yet DJs endorse the worn sound.

Alternative DJ Controllers

The product catalog of commercial physical controllers for DJing with other sound formats than vinyl is growing, and academic conferences such as New Interfaces for Musical Expression (NIME) have reported on innovative interfaces (for instance, Andersen 2003; Beamish, MacLean, and Fels 2004; Hansen and Bresin 2006; Lippit 2006; Fukuchi 2007; Pabst and Walk 2007; Villar et al. 2007; Hayafuchi and Suzuki 2008). New physical interfaces broadly fall into three categories: (1) interfaces converting the rotation speeds of traditional turntables to control signals, (2) interfaces using the turntable metaphor, such as with a rotating pad or jog wheel, and (3) interfaces using other metaphors, for example by interacting with graphical representations of sound files on touch-sensitive surfaces. Categories 1 and 2 are currently dominant in the commercial market.

In addition to the DJ control interfaces, sequencer-based interfaces with non-real-time interaction have also been developed, which enable users to compose scratch phrases, design new techniques, and play with different samples. For example, Scratcher (Faulstich 2007) is a Max/MSP patch for creating scratch phrases taken from a list or drawn in the TTM notation system (Turntablist Transcription Methodology; Carluccio, Imboden, and Pirtle 2000); and Auto-DJ (Wun, Yong, and Chan 2007) is an application for generating mobile-phone ringtones, using voice recordings manipulated by scratch techniques (*chirps* and *stabs*).

Skipproof combines features found both in real-time control interfaces and in sequencer-based interfaces. The input devices used to control the software can belong to any of the three aforementioned interface groups. An intuitive, easy-to-learn interface provides real-time control and manipulation of the most common scratch techniques as performed by DJs, so that the low-level, traditional techniques need not be practiced and mastered. The next section describes the Skipproof application, including its interface and mapping.

The Skipproof Application

The name and concept came from the "skip-proof" feature in specialized vinyl records, which was introduced by DJ Swamp (1998). In a skip-proof section of such a record, a sound with a duration corresponding to one rotation is repeated for a couple of minutes; thus, if the needle jumps out of its current groove in the record, it will probably stay in the same temporal location within the same sound sample.

Skipproof was developed in Pure Data (Pd, Puckette 1996), which facilitates control interfaces that use common data protocols such as Open Sound Control (OSC, Wright and Freed 1997), MIDI, USB, or TCP/IP. Pd was chosen as it is good for fast prototyping and easy collaborations.

A high-level *scratch preset* control mode and a low-level *scratch improvisation* control mode can be interchanged seamlessly during a performance. The improvisation mode, simulating the crossfader and turntable, is the default. The scratch preset mode is temporarily activated when the performer plays one of the modeled techniques.

Turntable Simulation

Skipproof provides the main functionality of a turntable and a mixer: the user can play different sound samples and alter the speed and amplitude manually. The typical controllers found on the traditional instrument—such as a start-stop button, a pitch slider for adjusting the motor speed continuously, revolutions-per-minute (RPM) selectors, and volume sliders—are emulated. For instance, a power-toggle and tap function can be used to simulate how DJs generate a slow, stepwise deceleration by turning off the motor while lightly tapping the vinyl. Similarly, motor-speed toggles for switching between 33 and 45 RPM can be used to simulate a vibrato-like effect that is created on turntables by pressing down the RPM selectors alternately.

The volume fader, which is used sparingly in scratching, was implemented as a linear amplitude control. Because the crossfader is traditionally used in a way resembling an on–off switch, and because scratching usually involves only one turntable, the crossfader was implemented as a logarithmic, triggered fading ramp going from sound to silence or vice versa. In Skipproof, it is thus not possible to fade between two sound sources or move the crossfader gradually. However, as on a scratch mixer, the steepness of the fading ramp can be adjusted, and the crossfader direction can be reversed by setting the end position to be muted instead of sounding (on mixers, this is called a "hamster switch").

Also, mechanical properties of the turntable were emulated and implemented as variable parameters; these include the pitch (speed change) amount, inertia, motor strength (torque), friction, and hand position on the record. Adjusting the inertia and torque will mainly have an effect on the time needed for the turntable platter to start up, slow down, and restore speed. The friction parameter can be adjusted to simulate how the friction between the vinyl and the platter is regulated by placing slipmats in between. The hand position parameter emulates the effect of adjusting the hand's distance from the record edge (where the same gesture generates shorter movements with a peripheral hand position than a central).

Turntables allow one to lift and move the needle between tracks while the record is spinning (so-called needle-dropping), and to physically mark the vinyl with (for example) adhesive stickers to quickly locate and cue the sound. Analogous features are available in Skipproof: Needle-dropping is possible as the sample can be changed without affecting the playing position, and cueing is emulated by a function that returns the sample to its start.

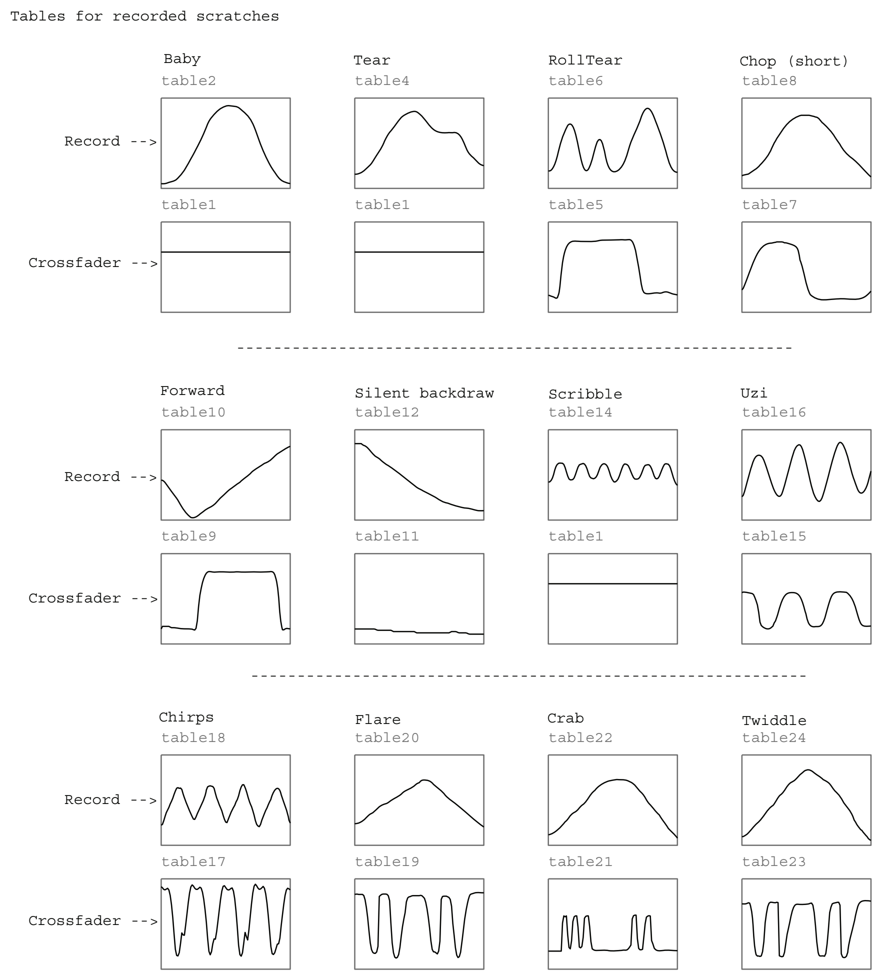
Scratch Techniques

The main feature of Skipproof is the implementation of twelve example scratch techniques, or "scratch presets." These are modeled based on the analysis of recordings by professional DJs, and they have been described in previous work (Hansen 2002; Hansen and Bresin 2003). The techniques included in the default list are *baby*, *tear*, *rolltear*, *chop*, *forward*, *silent back*, *scribble*, *uzi*, *chirp*, *flare*, *crab,* and *twiddle;* see Figure 3.

Skipproof has a default setting in which record gestures and the corresponding crossfader gestures are synchronized with each other. However, it is also possible to decouple the two gestures for creating new scratches and experimenting with them. This allows simulating a common performance practice where new techniques are created by changing the offset between the gestures (as can be observed in, for instance, DJ Q-bert 2007). In addition to changing the offset between gestures, the user can create new presets by sketching the record and crossfader movements in tables.

We will use the previously mentioned *chirp* technique as an example of how a scratch preset has been implemented. A professional DJ recorded isolated chirps, and chirps in sequences, using a turntable and mixer equipped with sensors to get the rotation speed and crossfader movement.

Figure 3. Scratch presets in Skipproof. The upper diagrams in each row are the record movement; the lower are the crossfader movement. In the baby, tear, *and* scribble *scratches, the crossfader is constantly on.*



The best chirps from the recordings were selected and extracted, and the sensor values stored in lookup-tables in Pd; see Figure 1c. The tables are resampled to have quantized durations and typical movement ranges (based on Hansen, Fabiani, and Bresin submitted). A reading of the table, triggered by the user, then sets the amplitude and deviation from nominal playback speed, thus generating a chirp scratch originating from the real DJ performance, but with properties that can be changed.

Although the speed and the extent of both the crossfader and the vinyl movements can be manipulated in the playback of the gesture, it was important to preserve the character of the technique: Most techniques, like the chirp, require precise coordination between record and crossfader in order to sound right. Although the parameters of a preset scratch can be changed, there is little possibility for real-time control during the playback; a scratch technique is typically shorter than 200 msec.

Certain scratch techniques like scribbles and chirps are usually played in sequences of the same gestures, whereas others like crabs and tears can be played singly. Chirps are therefore triggered in series in Skipproof; however, this series can be interrupted.

As an alternative to the recorded scratch presets, it is also possible to play idealized versions of the scratches, which are based on equations derived from interpolation of the gesture data from selected recordings. These models include both record and crossfader movements, and can be manipulated in the same way as the recorded presets.

Audio

To simulate skip-proof sections of DJ records, all sound samples that are used in Skipproof have a duration corresponding to one revolution on the turntable (1.8 sec at 33 RPM). Technically, however, sound files can have arbitrary length. As with a repeated groove on a record, the sound file is looped, so it is possible to scratch beyond the start and ending of the sound.

For a more realistic simulation of the vinyl record sound, the user can choose a level of wear, achieved by filtering the sound and adding hiss and click noises that are looped with the sample. Deteriorated record quality is also an important characteristic of the scratch sound, where the high-frequency content in the vinyl signal will be replaced by broadband noise (see Hansen and Bresin 2003 and Figure 2).

Even audio sources other than the Skipproof sound samples can be manipulated. For instance, the DJ gestures can control sound-synthesis parameters or—as in the case of the Reactable, explained shortly—another system with its own audio engine.

Sound quality in digital DJ tools has so far not been fully satisfactory; DJs particularly object to sound quality at low speeds. Also, even the slightest latency between the gesture and sound is reported as annoying. However, the main challenges for high-level control of the models are not necessarily latency and audio quality, but rather the controller and parameter mapping.

GUI and Visual Feedback

Skipproof's graphical user interface (GUI) was designed mainly for the purposes of testing the application and studying scratches. It was not intended to be used as a performance interface; however, it facilitates visual feedback when used in combination with hardware controllers.

Each function has been made available to the user through the GUI (see Figure 4). For instance, the gray rectangle in the interface represents the vinyl record, and moving the mouse in this area will affect the playback. In the default mapping, vertical movement changes the speed (and direction), while horizontal movement and position control the hand position and friction parameters, as well as parameters of added audio effects like echo (see bottom right in Figure 4). The vertical position does not correspond directly to the sound file position, as this mapping was considered by DJs to be less consistent with the idea of a rotating turntable. Conceptually, the user moves the record with the mouse (or other gesture controllers), and not the needle's position in the sample. The progress bar immediately to the left of the waveform display indicates the current playing position.

Compared to vinyl, digital audio makes adding new sounds straightforward, with all music available in a searchable library. However, when using computer interfaces, the immediate visual assessment of playback position that is available when using physical vinyl records will often be replaced by a visual indication that is now decoupled from the sound source and the place of the action.

Although new interfaces present new visualization possibilities, many musicians would naturally be reluctant to have to relearn their instrument. As Skipproof uses only short repeated sounds, visualization was not considered to be a critical issue.

Sensor and Parameter Mapping

Instruments intended for virtuoso playing, as most traditional acoustic instruments are, need to be predictable and endorse skillful handling; therefore they require low-level control. For example, a violinist must carefully regulate the bow force to achieve a desired amplitude. With computer-based instruments, however, it can be beneficial to give the player simultaneous control over several low-level parameters, as for instance in the case of playing a synthesized violin on a keyboard, where key velocity is mapped to amplitude—ranging from barely audible to loud—without allowing imperfect tones.

The low-level control-based instrument of concern here is, of course, the typical DJ equipment

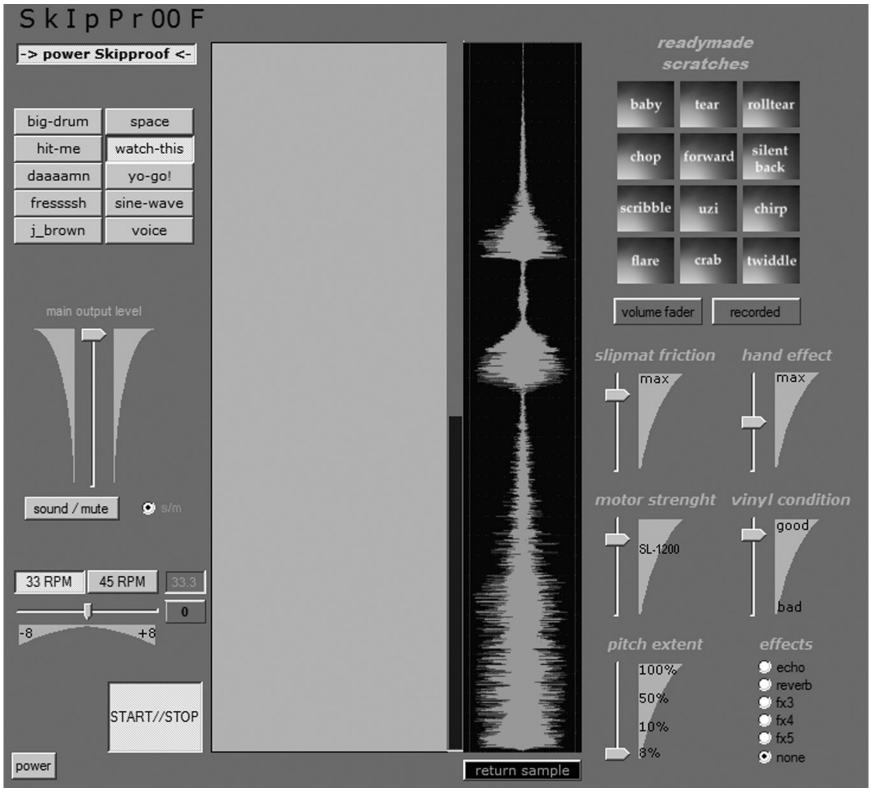


Figure 4. A screenshot of the Skipproof GUI. Typical instrument controllers are placed on the left side: motor power, sound samples, volume controls, pitch controls, and start button. On the right side are scratch technique presets, preset settings, the mechanical parameters such as friction, motor properties and record wear, and additional audio effects. The middle part has a scratch area (large gray rectangle), in which mouse motions change playback speed and direction, a vertical waveform display of the track, and, between these two, a vertical progress bar indicating the current playback position relative to the waveform display.

for scratching, and Table 1 shows the mappings between the most significant control parameters and sound parameters for this instrument. Instrument mapping for DJ scratching has been described earlier in Hansen and Bresin (2006); and in Hansen, Fabiani, and Bresin (submitted) it was found that for expressive performances, durations and onsets were varied more than (for instance) pitch. Timbre and dynamics were found to vary mainly because of other parameters that were being manipulated.

Control and sound parameters in Skipproof can be mapped either like the traditional DJ equipment or more unconventionally. For example, the pushing-and-pulling gesture has become a stereotypical trademark of DJs, but now we can explore other metaphors for changing the playback speed—for instance, by using pressure or light intensity as the control input.

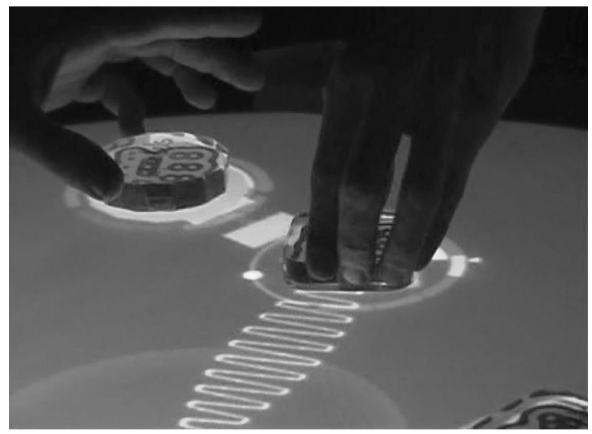
The mapping process in Skipproof was generalized in Mandoux and Wohlthat (2004). Here an easy connection and calibration of sensors was provided, with fast changing of the different mapping layers between controllers and sound parameters, following the schemes suggested by Hunt, Wanderley,

Table 1. Control and Sound Parameters for Scratching with a Physical Turntable and Mixer

	Duration	*Pitch*	*Onset*	*Dynamics*	*Timbre*
Record speed	●	●	◉	○	○
Sound sample	○	◉	●	○	◉
Sample position	●	○	●	○	○
Crossfader	●		●		
Volume fader	○		◉	○	
Tone control					○

● = The most important features; ◉ = Other common features; ○ = Features that are little used or implicitly controlled.

(a)



(b)

Figure 5. Performance situations with two different interfaces for controlling Skipproof: (a) Radio Baton with Skipproof GUI on a monitor, and close-up of finger transmitter and antenna; (b) the Reactable scratch objects played by two musicians, showing the virtual connection between objects.

and Paradis (2003). However, it is not trivial to find sensible mappings between sensors and the high number of control parameters available. This is especially true for high-level control, where the conceptual coupling is vague between (1) the simplified control action that generates a simulated scratch technique and (2) the combinations of gestures that are normally needed to create the same sound using physical equipment.

Skipproof Use Cases

In the following section, we present some use cases in which Skipproof has been played in combination with different interfaces and sensors. The musician's interaction and the resulting performances have been informally evaluated to various extents and for different purposes.

Scratching with the Radio Baton

The Radio Baton (Mathews 1991) is a gestural controller developed primarily for conducting of synthesized music sequences. A rectangular antenna registers the 3-D position of two transmitting batons held above its surface. The position data are processed to control musical events.

In the first public performance featuring the Skipproof software (see Figure 5a), the Radio Baton was used as the turntable controller. The player wore a transmitter fitted onto a finger, and movement above the antenna was mapped to the various parameters of the turntable simulation. As on a turntable, speed was controlled by horizontal movements on the y-axis. Vertical (z-axis) distance to the antenna was mapped to hand position on the vinyl, i.e., holding the hand close to the antenna corresponded to grabbing closer to the record center, resulting in more-effective hand movements. In addition, the x-axis hand position controlled audio effects such as echo.

In the first performance, the DJ used a standard mixer, but in later performances, crossfading was done by obstructing light sensed by a narrow-field optical sensor, allowing a nimble and precise control of onsets and durations. Conceptually, breaking a light beam could be compared to muting a sound, but the gesture did not resemble that of moving the crossfader. It was possible to perform fast techniques like the crab scratch (see Figure 3, bottom) easily by using the fingers to cross the light beam in a comb-like way.

High-level control was used in combination with the turntable simulation. The scratch techniques were triggered either by foot pedals or by gestures that approached defined areas on the antenna. The finger's speed and its height at the trigger point determined the speed and the extent, respectively, of the scratch.

Evaluation

The interaction was evaluated by the performing DJ during the development phase and also after a public performance. Mostly, the evaluation compared the hardware interface to the traditional instrument. It was considered that having no mechanical contact or feedback opened up new possibilities, especially with the two added dimensions (sideways and vertical hand position). Scratch preset playback was reported to be a welcome feature; it was used to a large extent during the performance.

Among the drawbacks of the Radio Baton interface, the performer mentioned latency, the fact that the finger-held transmitter had a wire, movement-detection imprecision, and the decoupling of visual feedback from the instrument. Regarding the Skipproof functionality, the lack of beat synchronization of the scratch techniques and the impossibility of setting a general tempo were indicated as problematic.

Skipproof on the Reactable

The Reactable is a circular tabletop musical instrument played by controlling optically tracked objects (Jordà et al. 2007). The physical objects positioned on top of the semi-translucent surface are linked to virtual objects that connect to each other. This connection is visualized with a rear-projected GUI, which in turn makes virtual connections between the real objects. In essence, the Reactable is a tangible performance interface for Pd.

Skipproof was integrated into the Reactable system as a set of new or modified Reactable objects, and it was tested with professional musicians. The Reactable scratch objects have been described in Hansen and Alonso (2008). In summary, the focus was on high-level control with three objects representing the sample, the record movement, and the crossfader movement, respectively. These objects could be manipulated, allowing the performer to control technique type, execution properties, and sample position.

Unlike the situation with the Radio Baton, the crossfader and record gestures were not synchronized. This permitted new combinations and techniques even with only a few basic movement patterns. The objects had durations corresponding to rhythmical beat subdivisions provided by a global metronome. The subdivisions and durations were controlled by the distance between connected objects and the distance from the center of the table, in compliance with expected Reactable behavior.

To experiment with low-level control, which is normally not the intended operation of the Reactable, "manual" sample and crossfader objects were designed. The sample object approximated an expected Reactable behavior where object rotation mapped directly to record movement, although here it was implemented as a temporary deviation from a nominal speed. When the crossfader object was moved, it would in one setting pass sound from the sample object and in another setting mute the sound from the sample player. This simulated the crossfader, but deviated from the expected Reactable behavior.

As expected, a video-based platform could not quite perform fast enough for low-level, real-time tasks. For high-level control, however, the latency introduced by the image processing was not critical.

Evaluation

Skipproof on the Reactable was tested with an expert Reactable player and an expert scratch DJ, who both did three practice sessions followed by questionnaire and interview evaluations (Hansen and Alonso 2008). Between sessions, the design was modified based on their comments. We found that their appreciations of the interface were diverging. Both rated high how they liked the instrument, but overall the Reactable player got more satisfied across sessions, whereas the DJ got less satisfied.

Already from the first meeting, the musicians could perform rather complex and realistic-sounding scratch patterns, and after a couple of hours of practice, they performed expressively. They judged that the instrument needed a lot of practice, and also that their performance did improve with practice.

Both were satisfied with the visual feedback. We found differences related to their field of expertise; only the DJ accepted non-standard Reactable behavior, whereas the Reactable expert was more forgiving of latency and movement inaccuracy. Both found that it got easier to perform scratching musically, predict the instrument behavior, and have more control, but also that it got harder to perform with the scratch technique objects over time. One reason for this could be that they performed increasingly more complex phrases as they practiced, but that the instrument could not provide a proper balance between their control of the interface and their musical ambition with the objects.

Scratching with a Friction Sound Model

Friction between two surfaces and scratching a record have several analogies: The scratching gesture is associated with the needle's sound from the friction against the vinyl, and the sound of scratching is arguably less "musical" than musical instrument sounds, but more similar to friction noise. These observations led to an experiment in controlling a friction sound model with Skipproof.

Serafin, Avanzini, and Rocchesso (2003) described a physics-based model of frictional interaction implemented in Pd. This model has many control parameters, such as force, velocity, and resonator and exciter characteristics, which all need to be carefully adjusted in correspondence to each other to provide the intended outcome, for instance to generate bowed string sounds.

In a recent project (Hansen, Alonso, and Dimitrov 2007), scratch techniques were used for controlling the friction sound model of Serafin, Avanzini, and Rocchesso (2003). The model was controlled both with the typical mapping where playback speed and crossfader control frequency and amplitude, respectively, and with alternative mappings to control (for instance) velocity and force parameters. A combined interface between Skipproof and the Reactable was tested to control the friction sound model within a specified functioning range to generate bowed string-like sounds (Dimitrov, Alonso, and Serafin 2008).

Evaluation

The interaction was informally evaluated by the authors and colleagues, and also by the Reactable expert and DJ mentioned earlier. Controlling the frequency and amplitude of friction sounds worked well, but attempting to tune the physical model to generate violin-like sounds with scratching was quite problematic. Thus, even if the interaction was found engaging, the musical output was not.

Other Examples of Using Skipproof

In addition to these examples, parts of Skipproof have been featured in other prototypes and applications. An ongoing project, *Ljudskrapan* [the Sound scraper], is aimed at helping children with cochlear implants and limited motor control to explore their hearing. Among other models, scratch techniques are used to manipulate sound samples in a playful way, producing a more complex output than what traditional musical instruments could.

Another interesting area of development is Skipproof-based applications for mobile devices, such as for creating ringtones (Wun, Yong, and Chan 2007) and for expressive interaction. Sancho (2009) recently developed an application for superimposing scratch techniques from Skipproof on mobile phone music players, which lets the user trigger scratch phrases but not perform improvised scratching.

Discussion

Through various experiments like those described herein, Skipproof has shown to be a versatile tool for working with scratch techniques. Skipproof includes only some of the basic techniques as presets, but because most techniques are variations on distinct record and crossfader gesture combinations, new models can easily be added. The Reactable setting showed that even with a limited subset of techniques available, the user can produce a variety of different scratches.

Experimenting with different interfaces and mappings makes it possible to study how both DJs and other players use scratch techniques expressively in performances. With the Radio Baton, simple gestures would trigger "perfect" scratches, but the player did not have much control of the continuous composition of techniques, as they were triggered individually. With the Reactable, it was easier to create richer scratch phrases as the objects would keep generating patterns while connected, and the player's effort could be focused on changing and manipulating the techniques. In this sense, the Reactable offered a more effective high-level control, which would suggest that DJ user interfaces do not necessarily have to resemble the traditional instrument.

When performing with sensors, we tried to minimize the visual feedback, but when using the Reactable, the visual feedback became an integrated part of the instrument. Even though we did not anticipate that visual feedback was very necessary for short looped sounds, it turned out that the DJs preferred such feedback as close to the source as possible.

Scratching is a topical case study for interface design, as DJs represent a large group of experts who interact expressively with hardware intended for a different purpose. Many scratch DJs develop ambidextrous skills, changing which hand controls the crossfader and which hand controls the turntable—this ability is imperative for certain DJ playing styles related to scratching, such as beat-juggling. Two-handed input is specifically mentioned as a future research direction in human/machine interaction (Buxton et al. 2002), and scratching can thus be an example of an interface where both hands independently perform highly trained actions, and where the users can perform the tasks with either the left or the right hand.

Because there are many new interfaces targeting DJs, it is worthwhile to reflect on which features could improve the possibilities for expressing musical ideas. A simulated turntable such as Skipproof can, for instance, be made to go beyond the physical constraints of the instrument. For instance, the scratch presets can easily produce unrealistically fast or slow scratches—which the DJs participating in our evaluation sessions gladly explored. The user can also manipulate sound sources other than Skipproof's internal audio samples, such as sound-synthesis parameters and external audio engines. However, the characteristic features of the instrument, such as physical restrictions and the vinyl audio quality, should be regarded.

In our close contact with the DJ community we have experienced that DJs are open to testing new interfaces, but they have high demands on low-level control of the instrument. Although scratch DJs may be perfectly content with existing interfaces, the higher-level control makes scratching sounds accessible to musicians who are eager to experiment but less devoted to learning the instrument. Also, high-level control of scratching can be useful in other musical applications such as sequencers, as is the case for most musical instruments.

Acknowledgments

The authors would like to thank the Vestax Corporation for equipment. Financial support was received in the early stages from the EU projects Sounding Objects, AGNULA and S2S[2] (Sound to Sense, Sense to Sound), and recently from the EU projects Braintuning (FP6-2004-NEST-PATH-028570) and SID (Sonic Interaction Design, European Cost action IC0601).

References

Andersen, T. H. 2003. "Mixxx: Towards Novel DJ Interfaces." *Proceedings of the Conference on New Interfaces for Musical Expression.* Montreal: Canada Faculty of Music, McGill University, pp. 30–35.

Beamish, T., K. MacLean, and S. Fels. 2004. "Manipulating Music: Multimodal Interaction for DJs." *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York: ACM, pp. 327–334.

Buxton, W., et al. 2002. "Human Input to Computer Systems: Theories, Techniques and Technology." Working draft, Available on-line at www.billbuxton.com/inputManuscript.html. Accessed 22 August 2009.

Carluccio, J., E. Imboden, and R. Pirtle. 2000. "Turntablist Transcription Methodology." Available on-line at www.ttmethod.com/. Accessed October 2009.

Dimitrov, S., M. Alonso, and S. Serafin. 2008. "Developing Block-Movement, Physical-Model Based Objects for the Reactable." *Proceedings of the Conference on New Interfaces for Musical Expression.* Genova: Infomus, Casa Paganini, pp. 211–214.

DJ Q-bert. 2003. *Do-It Yourself Vol. 1.* ThudRumble DVD: DIY001-DVD.

DJ Q-bert. 2007. *Scratchlopedia Breaktannica: 100 Secret Skratches.* ThudRumble DVD: SECRT001-DVD.

DJ Swamp. 1998. *The Skip-Proof Scratch Tool Volume 1.* LP: Decadent Records DEC002.

Faulstich, A. 2007. Scratcher. Software available on-line at www.skrasoft.com/. Accessed October 2009.

Fukuchi, K. 2007. "Multi-Track Scratch Player on a Multi-Touch Sensing Device." *Entertainment Computing – ICEC 2007*, *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Verlag, pp. 211–218.

Hansen, K. F. 2002. "The Basics of Scratching." *Journal of New Music Research* 31(4):357–365.

Hansen, K. F., and M. Alonso. 2008. "More DJ Techniques on the Reactable." *Proceedings of the Conference on New Interfaces for Musical Expression.* Genova: Infomus, Casa Paganini, pp. 207–210.

Hansen, K. F., M. Alonso, and S. Dimitrov. 2007. "Combining DJ Scratching, Tangible Interfaces and a Physics-Based Model of Friction Sounds." *Proceedings of the International Computer Music Conference*, Vol. 2. San Francisco, California: International Computer Music Association, pp. 45–48.

Hansen, K. F., and R. Bresin. 2003. "Complex Gestural Audio Control: The Case of Scratching." In D. Rocchesso and F. Fontana, eds. *The Sounding Object*. Florence: Mondo Estremo, pp. 221–269.

Hansen, K. F., and R. Bresin. 2006. "Mapping Strategies in DJ Scratching." *Proceedings of the Conference on New Interfaces for Musical Expression.* Paris: IRCAM, Centre Pompidou, pp. 188–191.

Hansen, K. F., M. Fabiani, and R. Bresin. Submitted. "Analysis of the Acoustics and Playing Strategies of Turntable Scratching." *Acta Acustica United with Acustica*.

Hayafuchi, K., and K. Suzuki. 2008. "MusicGlove: A Wearable Musical Controller for Massive Media Library." *Proceedings of the Conference on New Interfaces for Musical Expression.* Genova: Infomus, Casa Paganini, pp. 259–262.

Hunt, A., M. M. Wanderley, and M. Paradis. 2003. "The Importance of Parameter Mapping in Electronic Instrument Design." *Journal of New Music Research* 32(4):429–440.

Jordà, S., et al. 2007. "The ReacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces." *Proceedings of the International Conference on "Tangible and Embedded Interaction" (TEI07).* Baton Rouge, Louisiana: ACM, pp. 139–146.

Lippit, T. M. 2006. "Turntable Music in the Digital Era: Designing Alternative Tools for New Turntable Expression." *Proceedings of the Conference on New Interfaces for Musical Expression.* Paris: IRCAM, Centre Pompidou, pp. 71–74.

Mandoux, F., and S. Wohlthat. 2004. "Designing of a HCI for Skipproof." Technical report. Stockholm: Speech, Music and Hearing, KTH. Available on-line at www.speech.kth.se/~kjetil/files/Mandoux_Wohlthat_report.pdf.

Mathews, M. V. 1991. "The Radio Baton and Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music." *Computer Music Journal* 15(4):37–46.

Pabst, A., and R. Walk. 2007. "Augmenting a Rugged Standard DJ Turntable with a Tangible Interface for Music Browsing and Playback Manipulation." *Third IET International Conference on Intelligent Environments.* Ulm: Ulm University, pp. 533–535.

Puckette, M. 1996. "Pure Data: Another Integrated Computer Music Environment." *Proceedings of the International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 269–272.

Sancho, O. B. 2009. "DJ Tool: A Mobile Phone Audio Player Application." Master's thesis, KTH Computer Science and Communication, Stockholm.

Serafin, S., F. Avanzini, and D. Rocchesso. 2003. "Bowed String Simulation Using an Elasto-Plastic Friction Model." *Proceedings of the Stockholm Music Acoustics Conference*. Stockholm: Speech, Music and Hearing, KTH, pp. 95–98.

Sloly, D., and T. Frederikse. 2004. *Basic DJ Techniques*. London: Sanctuary Publishing.

Smith, S. 2006. *The Compositional Processes of UK Hip-Hop Turntable Teams*. Ph D thesis, De Montfort University.

Villar, N., et al. 2007. "The Colordex DJ System: A New Interface for Live Music Mixing." *Proceedings of the Conference on New Interfaces for Musical Expression.* New York: ACM, pp. 264–269.

Webber, S. 2007. *DJ Skills: The Essential Guide to Mixing and Scratching.* Oxford: Elsevier Science and Technology/Focal Press.

Wright, M., and A. Freed. 1997. "Open Sound Control: a New Protocol for Communicating with Sound Synthesizers." In *Proceedings of the International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 101–104.

Wun, S., C.-H. Yong, and T.-E. Chan. 2007. "Musical Extrapolation of Speech with Auto-DJ." In *Multimedia '07: Proceedings of the 15th International Conference on Multimedia.* New York: ACM, pp. 795–798.