Multi-track Scratch Player
On A Multi-Touch Sensing Device

Kentaro Fukuchi

Graduate School of Information Systems
The University of Electro-Communications
Choufu-shi, Tokyo , JAPAN 182-8585
fukuchi@megaui.net

Abstract. Scratching with turntables is a popular sound generation technique in today's music scene, especially in hip-hop culture. A conventional turntable system consists of two turntables (record players) and an audio mixer, but the proposed system requires a computer and a multi-touch sensing device, so it is smaller and portable. Moreover, the proposed system enables the use of various novel scratching techniques that are difficult or impossible to perform on existing systems. In this paper we describe the implementation of the proposed system and introduce some scratching techniques that can be used with it.

1 Background

Scratching on a turntable is a popular sound generation technique in today's music scene. Scratching usually involves the use of two turntables and an audio mixer for scratching. A player (known as a *DJ* or *turntablist*) manipulates these with his hands (Figure 1). A scratching sound is generated by rotating a record on the turntable back and forth by hand, while the other hand moves a cross fader or volume slider[7][8]. Therefore, both hands are busy in scratching so the records cannot be changed during a scratching motion and scratching two or more records is not possible. This limitation is considered to be an important problem in scratching, especially for the "no-tricks" style of scratching performance whereby a player performs music by scratching alone.

A simple solution to this problem is the use of more turntables, but they are too large (at least 12 inches square) to allow manipulation of multiple turntables by hand. There are a number of software-based scratching applications, but these employ conventional pointing devices or a dedicated external input device which has two rotary encoders to simulate a pair of turntables. They do not allow scratching of two or more soundtracks.

We developed a computer-aided scratching system that enables simultaneous multi-track scratching by using a multi-touch sensing device. The system is small and portable, and enables the use of various novel scratching techniques that are difficult or impossible to perform on existing systems.

2 Related Work

Audiopad[11] allows multitrack audio mixing by Phicons that contain RFID tags. By attaching a soundtrack to a Phicon and moving them, the user can control the volumes of the soundtracks concurrently, but scratching is not possible.

Fig. 1. Scratching using a turntable

KORG KAOSS PAD[10] is a commercial product that allows scratching by moving a finger on an embedded touch sensor. The sensor does not detect multiple touches so it is not possible to scratch multiple soundtracks.

Skipproof[6] allows a scratch performance with various input devices such as MIDI interfaces or gesture sensors. The user can scratch a single 1.8 second long audio sample with an input device. In addition, the user can trigger a pre-recorded (readymade) scratching technique instead of manual scratching.

Jeff Han introduced a multi-touch sensing technique that utilizes an internal reflection in a clear acrylic board[5]. They also developed a software-based musical application on it.

3 System Architecture

3.1 Multi-touch Sensing Device

We employed SmartSkin[13], a human-body sensing device based on capacitive sensing. The sensor recognizes multiple hand or finger positions and shapes, and calculates the distance between the hand and the surface. We used a tablet-sized SmartSkin that has 32×24 grid cells and each cell is 9 mm×9 mm.

We used a pair of tablet-sized SmartSkin sensors to create a larger input surface that is 47 cm×31 cm. Both SmartSkins are connected to a computer via USB 1.1, and the computer estimates the motion of the fingertips from the sensor data from the input data then generates sound. The resolution of finger positions is 1504×992.

3.2 Display

A projector is located above the input surface, and the projector displays the screen on the surface. The resolution of the screen is 1504×992, which is the same as the resolution of the input system. Figure 2 shows an overview of the system.

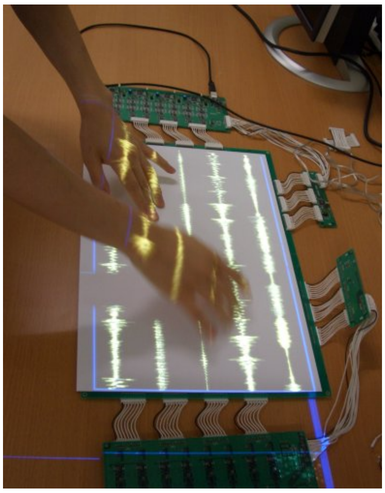

Fig. 2. An overview of the multi-track scratch player.

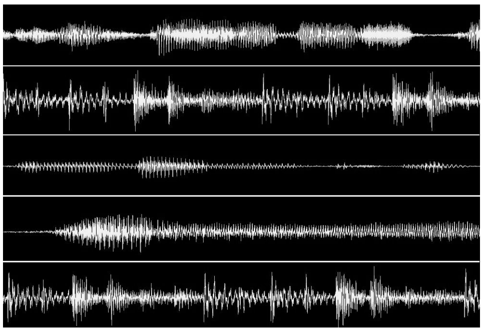

Fig. 3. Screenshot of the multi-track scratch player. Five soundtracks are shown, and each waveform is displayed horizontally. The performer touches the waveform with his fingers directly and performs scratching. The speed and direction of playback is synchronized to the motion of the finger.

3.3 Finger tracking

A multipoint input system was implemented on SmartSkin[4][3]. With this system, the user touches an input surface by his fingers directly. The sensing range of tablet-sized SmartSkin is 5 mm, which means that the system can distinguish whether or not a finger is touching the surface. As a result, a touch-panel-like multipoint input surface was developed. In this section, we describe the algorithm of the finger tracking.

First, the system scales up the sensor values by bicubic convolution interpolation. As shown in Figure 4, fingertips can clearly be seen in the interpolated image (bottom right). The system then detects the fingertips from the interpolated values. The values below a threshold are omitted to detect only contact or very close presence to the input surface parts. The bottom right section of Figure 4 shows a filtered image. Next, the system divides the values into several continuous regions by using the segmentation algorithm. If the number of pixels of a region is within a certain range, the system recognizes the region as a fingertip and calculates its centroid as the position of the fingertip.

SmartSkin cannot identify which finger of whose hand is on the grid. Therefore, in order to track the motions of the fingers, the system has to estimate the motions by comparing the current positions of fingers and their previous positions. A cost minimization analysis technique was employed for motion tracking. The details are as follows.

In frame t, let $F_{i,t}(i = 1, 2, \ldots, n_t)$ be positions of detected fingertips. When $F_{j,t+1}(j = 1, 2, \ldots, n_{t+1})$ is given as their positions in the next frame $t+1$, the system calculates a *candidate set* of corresponding points $S_{i,t}$ for every $F_{i,t}$: when a cut-off value R is given, $S_{i,t}$ consists of points of $F_{j,t+1}$ which satisfies $|F_{i,t} - F_{j,t+1}| < R$.

In the same way, the system calculates $S_{j,t+1}$ for $F_{j,t+1}$, which consists of points of $F_{i,t}$, which satisfies $|F_{i,t} - F_{j,t+1}| < R$. If $F_{i,t}$ has no candidate ($S_{i,t} = \emptyset$), it

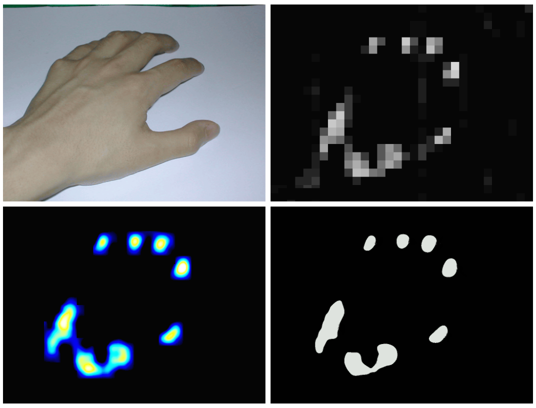

Fig. 4. Step of fingertip detection: A hand on the SmartSkin (top left), sensor values (top right), interpolated values (bottom left), after the segmentation process (bottom right).

is considered that the corresponding finger is released, and it is eliminated from the remaining process. If $F_{j,t}$ has no candidate ($S_{j,t+1} = \emptyset$), it is considered that the finger is newly detected, and it is eliminated as well.

Next, in order to determine a corresponding point for the rest of $F_{i,t}$, the system checks all of the combinations. Let a matrix T be a combination of $F_{i,t}$ and $F_{j,t+1}$. Its cost is calculated by following equation:

$$\sum_{i,j} T_{i,j} |F_{i,t} - F_{j,t+1}|^2$$

Then, the system compares all of the costs of T and choose the minimum cost. If the number of combinations is too large, the system aborts the tracking and treats all of fingers as being released. In this implementation, the cut-off value R is 200 pixels, and the limit of the number of combinations is 10^6.

Finally, the system uses a Kalman filter to absorb small shaking. However, because the scan rate of SmartSkin is as slow as per 20 scan/sec, its response became worse when the filter was applied naively. In order to avoid this, the system does not apply the filter when the motion length is longer than five pixels. In addition, the system does not use a Kalman filter to estimate the motion.

Figure 5 shows an increase in computational time for finger tracking. The computational time is averaged over frames of every number of fingers. The light gray area represents the time required to read the sensor values and interpolate them. The dark gray are represents the time required to detect and track the fingers from the interpolated data. As shown in the graph, the current implementation is sufficiently fast for one or two users for real-time processing.

3.4 Multi-track scratching

Figure 3 shows a screenshot of the screen that is overlaid on the input surface, as shown in Figure 2. Current implementation enables five sound tracks to be played, and the

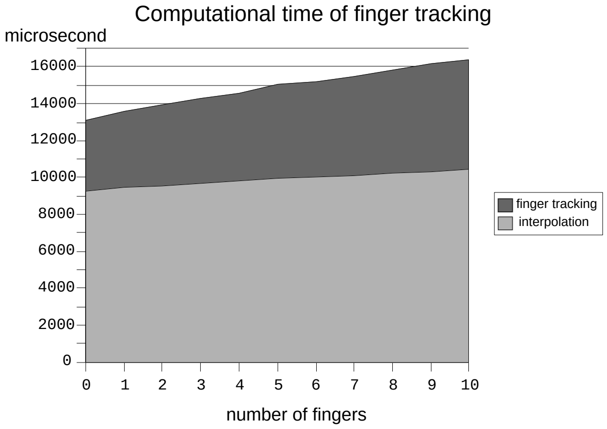

Fig. 5. Average computational time of finger tracking

sound wave from each track is shown horizontally. Each sound track is the same length as the PCM wave data.

If a track is not touched by a finger, the track is not played. When it is touched by a finger, the application plays the waveform that the finger passes over, as if the finger was a needle and the soundtrack was the groove of a vinyl record. The vertical motion of the finger controls the volume of the sound (optional for advanced use). This manipulation can be performed in parallel on all sound tracks, and the result is mixed down to the computer's audio output. Even when a sound track is touched by two or more fingers, it acts in the same manner.

3.5 Audio processing

The motion of a finger is used to make a scratching sound. When a new finger position is acquired, the application plays the waveform that the finger passes over within the interval of scanning.

The length of each soundtrack is 2–3 seconds. In order to play a soundtrack at normal speed, it is necessary to move a finger at 16–28 cm/s, because the width of the input surface is 47cm. The sampling rate of soundtracks is 44.1KHz. Therefore, a soundtrack contains 88200–132300 samples and a horizontal step of finger motion plays 88 samples at most. In order to play the waveform within a certain time, the waveform data must be resampled. At present, the application resamples the data discretely, which results in loss of the sound quality. Several techniques by which to avoid quality loss are known, such as Sinc-based resampling[14][1]. However, these techniques were not tested on our system.

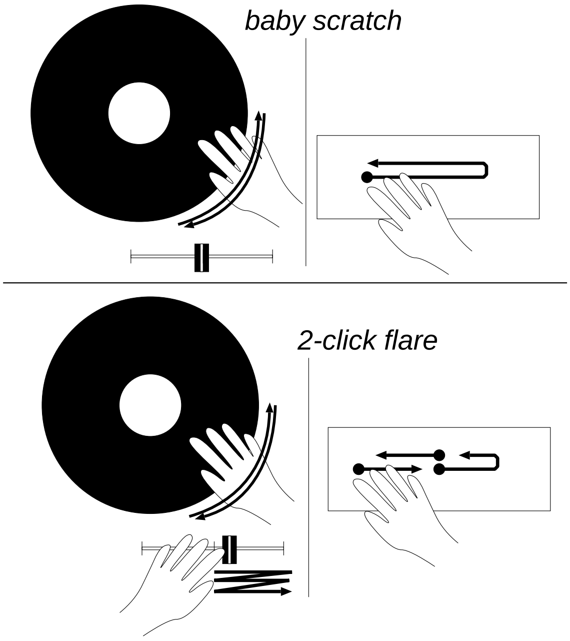

Fig. 6. Conventional scratching techniques on the Multi-track Scratch Player

4 Simulation of known scratching techniques

There are several scratching techniques for conventional turntables and audio mixers. A scratching technique consists of a pattern of motion of a turntable and a cross fader. DJs and turntablists use these techniques sequentially in a performance.

As well as enabling the use of known scratching techniques, the proposed scratching system is also designed to allow novel scratching techniques that cannot be performed on conventional systems.

Figure 6 shows two major scratching techniques on a turntable and the corresponding procedures for the proposed system. The technique shown at the top of the figure is known as the “baby scratch”. This is the simplest type of scratching, in which the player rotates the turntable back and forth repeatedly. This scratching can be performed by repeatedly moving a finger on a track horizontally.

The bottom figure shows the “two-click flare” technique, which requires concurrent manipulation of the turntable and cross fader. The player rotates the turntable back and forth using one hand, while the other hand quickly slides the cross fader between the min and max positions twice. By this fading motion, the player can cut the sound to make beats. In order to simulate this scratching technique in the proposed system, the player must release the finger from the input surface twice while moving his finger repeatedly. However, it is difficult to perform this motion quickly. To perform a rapid click, the player can play the first click with one hand then the second click with the other hand.

5 Discussion

This system was tested and a sound delay was found to be a significant problem. The SmartSkin sensor scans at 20 Hz but because the application plays a soundtrack between the current and previous positions of a finger, a 50 millisecond playback delay is generated. In music performance, this delay is unacceptable. In order to avoid this problem, we plan to alter the multi-touch sensing device such as iGesturePad[9].

Generally, a turntable rotates at a constant speed when a player does not touch it. The speed can be controlled by a slide volume. Moreover, it can be slowed down by touching the surface of the record softly, or increased by pushing it in the direction of rotation. At this time, the proposed system does not support this kind of manipulation. Some DJs have reported that these manipulations are very important for scratching. Therefore, in order to realize these existing scratching techniques, these manipulations should be supported.

Turntablists use a cross fader to click the sound (see Figure 6). In order to simulate this technique, our system requires the performer to move his fingers up and down very quickly, and this is not as easy as the fader action. We plan to add fader widgets next to the tracks. This idea is inspired by Vestax QFO[12], a turntable system with a built-in mixer, designed by a professional turntablist, Q-BERT.

6 Future Work

We have developed a multiple pointing input system based on camera-tracked laser pointers[2]. This input system allows users to point to multiple positions on a screen using laser pointers. We plan to implement the multi-track scratch player on the laser-based input system. In this case, we hypothesize that the problem of latency will be reduced because the laser based system scans positions at 30 scans per second. In addition it allows multi-user interaction of scratch playing.

We plan to use low-latency input devices and evaluate our multi-track scratch player to prove that our approach is useful for advanced DJ-style music. To evaluate these devices, we are using a high-speed camera to analyze the motion of a scratch play.

7 Conclusion

We developed a software scratching system with a multi-touch sensing device, which enabled concurrent scratching of multiple soundtracks. The proposed system allows some new scratching techniques as well as conventional ones to be used on turntables.

At this time, the proposed implementation causes a 50 millisecond delay, which is unacceptable for musical performance.

The proposed approach was well-received by DJs. Therefore, the scan rate of the multi-touch input device should be improved and the delay should be decreased. Finally, more new scratching techniques should be developed for the system.

References

1. Secret Rabbit Code. `http://www.mega-nerd.com/SRC/`.
2. K. Fukuchi. A Laser Pointer/Laser Trails Tracking System for Visual Performance. In *Human-Computer Interaction - INTERACT 2005*, volume 3585 of *Lecture Notes in Computer Science*, pages 1050–1053. Springer, 2005.
3. K. Fukuchi. *Concurrent Manipulation of Multiple Components on Graphical User Interface*. PhD thesis, Tokyo Institute of Technology, 2006.
4. K. Fukuchi and J. Rekimoto. Interaction Techniques for SmartSkin. In *Proceedings of UIST'02*, 2002.
5. J. Y. Han. Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM Press, 2005.
6. K. F. Hansen and R. Bresin. *The Sounding Object*, chapter Complex Gestural Audio Control: The Case of Scratching, pages 221–269. Mondo Estremo, 2003.
7. K. F. Hansen and R. Bresin. Analysis of a genuine scratch performance. In *Gesture-Based Communication in Human-Computer Interaction*, volume 2915/2004 of *Lecture Notes in Computer Science*, pages 519–528. Springer Berlin / Heidelberg, 2004.
8. K. F. Hansen and R. Bresin. Mapping strategies in dj scratching. In *Proceedings of NIME 2006*, pages 188–191, 2006.
9. iGesturePad. `http://www.fingerworks.com/igesture.html`.
10. KORG KAOSS PAD. `http://www.korg.co.jp/Product/Dance/KPE1/`.
11. J. Patten, B. Recht, and H. Ishii. Audiopad: A Tag-based Interface for Musical Performance. In *Proceedings of the 2002 International Conference on New Interfacefor Musical Expression (NIME02)*, pages 11–16, 2003.
12. QFO. `http://www.vestax.com/v/products/players/qfo.html`.
13. J. Rekimoto. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of CHI2002*, pages 113–120, 2002.
14. J. O. Smith and P. Gossett. A flexible sampling-rate conversion method. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 112–115, 1984.