Twkyr: a Multitouch Waveform Looper

Karl Yerkes
Media Arts and Technology
University of California Santa Barbara
yerkes@mat.ucsb.edu

Matthew Wright
CREATE & Media Arts and Technology
University of California Santa Barbara
matt@create.ucsb.edu

ABSTRACT

Twkyr is a new interface for musical expression that emphasizes realtime manipulation, audification, and visualization of waveforms with a multitouch display, offering different interactivity at different time scales, within the same waveform. The interactive audiovisual design of Twkyr is motivated by the need for increased parsimony and transparency in electronic musical instruments and draws from the the work of Curtis Roads on time scales as qualitative musical parameters and Edward Tufte's "data-ink" principles for the improvement of data graphics.

Keywords

ensemble, iOS, waveform, granulator

1. INTRODUCTION

Twkyr (shown in Figures 1 and 2) is a multitouch waveform-looping instrument. It presents the user little more than a waveform that can be scrolled, zoomed, and looped. With a sequence of simple gestures, the user may quickly adjust the time-scale of the display (zoom) and select sections of the waveform to play (loop). Although the metaphor that Twkyr presents is straightforward and consistent, the perceptual qualitative results vary greatly as a function of the time-scale of loop and often also depend on the nature of the sound material.

Twkyr's interactive audiovisual design is informed by the work of Curtis Roads on the perceptual qualitative differences between musical time scales and Edward Tufte's "data-ink" principles for the improvement of data graphics.

"As sound passes from one time scale to another it crosses perceptual boundaries. It seems to change quality. This is because human perception processes each time scale differently. Consider a simple sinusoid transposed to various time scales (1 µsec, 1 ms, 1 sec, 1 minute, 1 hour). The waveform is identical, but one would have difficulty classifying these auditory experiences in the same family" (Curtis Roads [3, p. 4]). Twkyr operates in four of Roads' nine time scales: "*Meso*: Divisions of form. Groupings of sound objects into hierarchies of phrase structures of various sizes, measured in minutes or seconds. *Sound object*: A basic unit of musical structure, generalizing the traditional concept of note to include complex and mutating sound events on a time scale ranging from a fraction of a second to several seconds. *Micro*: Sound particles on a time scale that extends down to the threshold of auditory perception (measured in thousandths of a second or milliseconds). *Sample*: The atomic level of digital audio systems: individual binary samples or numerical amplitude values, one following another at a fixed time interval. The period between samples is measured in millionths of a second (microseconds)."

Tufte's notion of a data graphic's "data-ink" is the "non-redundant ink arranged in response to variation in the numbers represented" [5, p. 93]. "Five principles in the theory of data graphics produce substantial changes in graphical design: (1) Above all else show the data (2) Maximize the data-ink ratio (3) Erase non-data-ink (4) Erase redundant data-ink (5) Revise and edit" [5, p. 105].

Twkyr is designed to devote all of the "ink" of the graphical display to representing the relevant numbers. In addition to the waveform, tracked touch positions, loop points, and playback cursor, Twkyr draws "frets" that help the user play specific pitches at the Micro/Sound-Object time scales and play rhythms at the Sound-Object/Meso time scales. Although the rules of waveform display and interaction are the same at all time scales, Twkyr subtly alters the visualizations (see Section 2.1.5) and/or the signal processing (see Section 2.4) according to the current time scales of both zoom level and loop size.

2. TWKYR

Audio editor Cool Edit Pro (of the late 90s) afforded zooming, scrolling, and looping of waveforms. With practice, one could use it as a sampler-like instrument to produce tones, rhythmic loops, phrases, or backing loops depending on timescale. The interface was clearly not meant for this; its modal, WIMP paradigm required careful timing and interplay between keyboard and mouse to accomplish these musical maneuvers. What if this interface were reimagined as a multitouch, modeless interface for performance? How high a ceiling and how low a floor could we get from just 1-finger zooming and scrolling and 2-finger looping? These are the questions that Twkyr poses.

2.1 User Interface & Interaction Behavior

See `https://vimeo.com/twkyr/demo` for videos of Twkyr.

2.1.1 No Interaction / Start Up

On start up and when there is no interaction, Twkyr presents three user interface elements: a fullscreen line-drawn waveform, the name of the sound file in the upper left-hand corner, and a thin grey box outlining the touchable area of the display. These elements are present at all times.

2.1.2 One-finger Interaction: Navigation

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

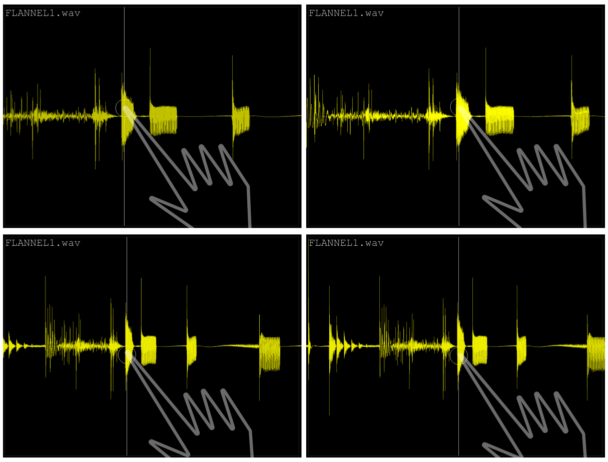

Figure 1: One-finger interaction: pan and zoom. All four images use the same underlying waveform; the superimposed translucent hand illustrates interaction and is not part of Twkyr's graphical output. The horizontal axis shows scrolling: the user touch moves horizontally and the waveform follows the touch. The vertical axis shows zoom: the user's finger moves vertically to change the time-scale of the display. In this figure motions are small relative to Twkyr's range, for visual clarity comparing waveforms.

Touching the display with a single finger causes Twkyr to draw a small circle at the finger position and a vertical line through that position. Moving this finger about the display navigates the waveform as shown in Figure 1. Horizontal movement scrolls by "dragging" the waveform left or right, as one might drag a piece of paper across a desk. When the finger first touches the surface, the timepoint at the horizontal position of the touch "locks" to the finger position: as long as the finger is held down, motion in the X (time) axis causes the waveform to scroll so that the originally-touched point always stays beneath the finger. The entire waveform is treated as a large loop (such that the first sample comes again immediately after the last sample), so this scrolling can never go beyond the ends of the waveform.

The finger's vertical movement zooms into and out of the waveform, with the zoom again centered around the waveform time point that is "locked" to the finger position, as with scroll. This mapping was inspired by Cool Edit Pro's use of the mouse scroll wheel, an otherwise free continuous controller (typically used in text document applications and web browsers to move up and down in a page), for facile, precise control of zoom over a wide range.

Three quick single-finger taps cause Twkyr to advance through a circular list of sound files.

2.1.3 Two-finger Interaction: Making Sound

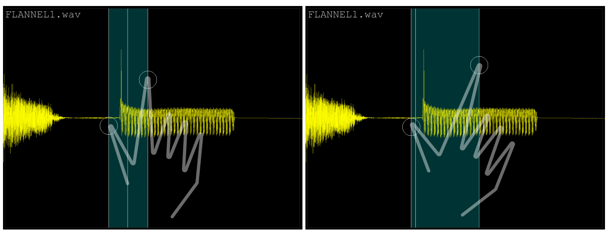

Figure 2: Two-finger interaction: loop selection.

Touching the display with 2 fingers "selects" the waveform region horizontally between them and plays it as a loop (Figure 2). At the moment of the second touch, playback begins from the position of the first touch towards the second (so that playback will be reversed when the first touch is to the right of the second). Releasing either finger immediately ends playback. The same small circle and vertical line visually represents both finger positions. The selection is filled in with light grey and the current "playback head" position is represented by a vertical line moving between the 2 fingers. Navigation is disabled.

Loop selection thus depends only on the X position of each finger, not Y. In the same spirit of non-wastefulness behind Tufte's "data-ink" principle, we map these free valuators to perhaps the most musically useful parameters: amplitude and frequency. The order in which the two fingers contact the surface determines the mapping of vertical position. The first finger down ("primary") controls amplitude via a logarithmic mapping: top is 6 dB, bottom is -48 dB. The second finger down ("secondary") controls the playback rate of the sound (± 1 octave).

2.1.4 Three-finger Interaction

While looping with two fingers, a third finger touch (which Twkyr visually represents by just a small circle) navigates much like a single finger does. We chose this symmetry to reduce the cognitive overhead on end-users, making the third mode understandable in terms of the other two.

However, musical demands lead us to introduce one small inconsistency: instead of zooming about the third finger, zooming is centered on the middle of the loop. This allows for more gradual changes in loop selection, particularly in the case where the third finger is outside the current loop.

2.1.5 "Frets"

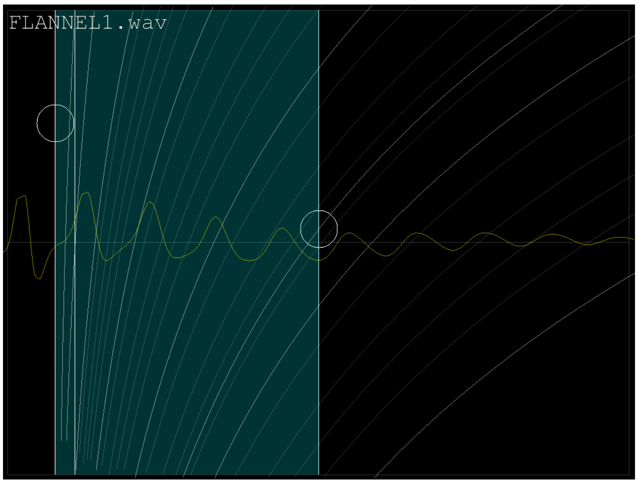

Figure 3: Screenshot of Twkyr with "frets" indicating an A minor pentatonic scale.

In Micro time-scales, where the loop frequency generally maps to perceived pitch, Twkyr overlays visual guides on the waveform as shown in Figure 3. Each of these "frets" is a curved contour line showing all positions for the secondary finger that will produce the same loop frequency, given the current position of the primary finger.

The curved shape of these "frets" comes from the fact that the frequency of the loop is inversely proportional to the length of the loop on the horizontal axis, but varies exponentially according to the $\pm$ 1 octave pitch control on the vertical axis, as shown in Equation 1.

Table 1: Visual Feedback at Various Time Scales of Zoom

Time Scale	*Frequency (Hz)*	*Samples Per Pixel*	*Twkyr's Behavior*
90.703us–2.9025ms	11025–344.53	0.00391–0.125	Show pitch frets. Show samples. Spline curve.
2.9025ms–23.220ms	344.53–43.066	0.125–1	Show pitch frets. Lines.
23.220ms–	43.066–	1–	Min/Max Overviews.

2.2 Gestures

Although the rules governing interaction are simple, they afford a variety of performance techniques in terms of the way musicians can control them over time both discretely and continuously, including one- and two-hand usage. The following Twkyr "idioms" were discovered by the developers and taught to the participants in our user study and in ensemble performances (see Section 2.5). The first step for every Twkyr gesture is framing: the user must first zoom to an appropriate time-scale and/or scroll to an area with useful sound material.

2.2.1 *Fade in a sound*

Place a single finger on the display, near the bottom. Place a second finger somewhere else on the display. Slowly slide the first finger up, or move it rapidly up and down for tremelo.

2.2.2 *Play in reverse*

Place a single finger on the display, then place a second finger to the left of the first finger.

2.2.3 *Play a percussive event*

Place a single finger on the display, then tap the display with a second finger and release quickly (before the sound can loop). To produce a decaying envelope, find an area of the waveform with those dynamics.

2.2.4 *Play with vibrato*

Use a single finger to zoom to a pitch regime time scale (see Table 1); curved "frets" appear. Place a second finger on the display, following the curves to stay "in tune" if desired. Wiggle the secondary finger on the surface without removing it to produce vibrato.

2.2.5 *Play in "Bagpipe" mode*

In a Micro time-scale make a very small (1–10 pixels) selection with 2 fingers. Vary the selection size in this range. As with bagpipes, the amplitude will remain fairly constant, with articulation provided only by discrete pitch jumps (among a subharmonic series, as described in Section 2.3.3).

2.2.6 *"Scrub" through a sound*

In a Meso time scale, make a small selection. Move both fingers horizontally across the display. The sound continuously looping between the moving endpoints will produce a sort of granular scrubbing effect following the fingers. Alternatively, hold the selection stationary while using a third finger to scroll the waveform under the selection.

2.2.7 *Play a "Hard sync"-like Effect*

Find a place in the waveform with pitched material. Zoom into a Micro time scale and make a selection.

2.3 Limitations & Affordances of the iPad

We choose to use the iPad for Twkyr because of its ubiquity and because it offers an integrated display and multi-touch interaction that may provide the user the impression of "touching the waveform."

The iPad 1, iPad 2, and iPad Retina Display are all approximately 240 × 190 mm in physical size, near the maximum span of fingers on most people's hands. As Figures 1 and 3 show, Twkyr is designed to be used in landscape orientation. We will refer to the horizontal and vertical axes as x and y. The iPad 1 and 2 have 1024×768 pixel displays, while the iPad Retina display has 2048×1536 pixels.

2.3.1 *Touch Resolution and Unreachable Pixels*

With a touchable display, generally the most useful representation for touch position is with the units and coordinate system of the visual pixels, so that developers can relate touch positions to the graphics being shown. This is what comes out of the Cocoa Touch frameworks that Apple provides to iOS developers.

Our inconclusive tests using two iPad 1 devices suggest that in general each display pixel is addressable with a single finger, implying that the actual resolution of the touch sensor is at least as good as the pixel resolution. The exception is near the edges, where there is a small, ragged margin of 0–2 outermost pixels that are never returned as touch position. Also, because the front face of the device extends beyond the boundaries of the touchable display, fingers might be outside the area where touches are sensed. Therefore, to prevent users from unintentionally stopping a loop when their fingers move too close to the edge, Twkyr renders a thin line 8 pixels from the edge of the iPad, encouraging users to play inside the box.

2.3.2 *Sub-pixel Touch Accuracy?*

Increased sensor accuracy is almost always a good thing for NIMEs, helping afford intimate musical control [6]. There is a low-level struct (GSEvent.h) reachable from one of Apple's "private frameworks" that may provide access to higher resolution touch data. As future work we would like to experiment with using higher resolution touch position.

The core of Twkyr's interface metaphor is the idea of directly grabbing a selection of a displayed waveform. In this context, the question of whether to quantize touch position to the nearest visual pixel is equivalent to the question of whether the user is selecting a region of *pixels* that display the waveform or selecting a more arbitrary region of *time* in the underlying waveform that the pixels coarsely display.

2.3.3 *Musical Effect of Touch Position Quantization*

The current version of Twkyr accepts the quantization of touch position to integer visual pixels, resulting in a very specific perceptual result when the selection points are very close: the subharmonic series. For example, suppose the user has zoomed in so that the number of samples per pixel $spp = 2$ and keeps the secondary finger at the vertical center of the iPad (so that there is no transposition). A single-pixel loop will create 22050 Hz, a two-pixel selection will produce 11025 Hz, a three-pixel selection will produce 7350 Hz, etc. No matter the zoom level, the periods of the shortest possible loops will form the series $spp, 2{\cdot}spp, 3{\cdot}spp, \cdots$

In general, these frequencies are available to the user:

$$f(\Delta x, y) = \frac{2^{(yMax/2-y)/yMax} \cdot sr}{\Delta x \cdot spp} \tag{1}$$

where Δx is the horizontal distance (in pixels) between the fingers and y is the vertical position of the secondary finger, $yMax$ is the display height and sr is sample rate. For the

Table 2: Effect of Loop Seam Discontinuity for Various Time Scales of Loop Length

Loop Length	Frequency(Hz)	Roads Time Scale [3]	Effect on Sound	Twkyr's Behavior
2 samples–23ms	11025–43	Micro–short Sound object	Changes timbre	Don't "fix" click.
23ms–92ms	43–11	Short sound object	Increasingly distracting click	gradual cross-fade
>92ms	<11	Sound object–Meso	Distracting click	"Fix" / cross-fade.

iPad 1's 1024×768 display, $\Delta x \in (1, 1023)$ and $y \in (0, 767)$, though in practice (Section 2.3.1) typically $y \in (3, 765)$.

This imposition of the subharmonic series has little effect for large-number-of-pixel loops because in these ranges small changes in finger position smoothly map to small changes in loop frequency, whereas small changes to small-number-of-pixel selections tend to produce a very specific musical character.

2.4 Time-Scale-Aware Behaviors

Twkyr presents a consistent interface while honoring qualitative perceptual differences among time scales. One example is that the "frets" (Section 2.1.5) appear only at the Micro time scale where they are perceptually meaningful.

Similarly, Twkyr's loop seam "click" repair behavior depends on time scale. Whenever the playback head crosses the loop seam there is most likely a discontinuity in the audio stream as the playhead jumps from the last sample in the loop to the first. The perceived musical effect of such a discontinuity depends on the length of the loop as well as the nature of the underlying material. For instance, if the loop length is in the Meso to longer Sound object time scale, then a discontinuity will likely be perceived as a disruption of musical meter and/or "click", whereas if the loop length is in the short Sound object to Micro time scale a discontinuity may be perceived as a bright timbre (See Table 2).

Therefore Twkyr "fixes" loop seam discontinuities for loops with lengths longer than 23ms by cross-fading across the loop seam. Starting at 23ms we gradually increase the cross-fade time linearly from zero until it reaches 10ms when the loop length is 92ms. For loops with length shorter than 23ms, we allow these discontinuities because they cause "hard sync"-like effects (Section 2.2.7) that can quickly and dramatically affect timbre.

2.5 User Evaluation

In an informal user study we taught 11 participants to use Twkyr and asked for feedback and feature requests. We then asked a group of experimental musicians to share their experiences practicing and performing with Twkyr.

Some users complained that the interface does not use the well-known iOS gestures such as pinch-zooming and flicking. However, after walking them through some Twkyr gestures they quickly adapted to Twkyr's different semantics.

Along with praise (e.g., "It is very tempting and fun to go crazy on this instrument!", "It's an instrument that offers a long term relationship", "Very flexible and responsive, with the right samples it felt pretty expressive."), several users requested "obvious" features (live sampling, easy loading of user waveforms, snap-to frets) already planned for development, as well as features we hadn't considered ("An option to control a release parameter of the envelope of sound", "the option to work with multiple loop heads", and "visual feedback" on what other ensemble performers are playing).

One performer stated that "its main drawback of sucking the user into staring at the screen makes it difficult for interaction [with other performers and the audience]."

Several participants spoke to the benefits of learning details of particular waveforms so as to gain facility performing that specific waveform. "it would be difficult to be an all-purpose Twkyr performer—it's much easier to learn a particular waveform well."

3. RELATED WORK / PRIOR ART

Samplr [1] offers a Twykr-like 2-finger looping gesture. However, it devotes half of its display area to buttons and sliders, greatly reducing the gestural space for direct waveform interaction and lowering the "data-ink" ratio. Also, Samplr does not offer waveform zooming.

Curtis [4] is a granular synthesizer that offers loop-based scrubbing through waveforms. Curtis uses a single finger to play loops, mapping vertical finger position to loop length to and horizontal finger position to loop source material. It does not offer zooming and devotes half of its display area to buttons and sliders leading to a low "data-ink" ratio.

Borderlands [2], a minimalist waveform granulator, devotes zero display area to buttons and sliders while embracing well-known iOS gestures like pinch-zoom, flick, rotate, and double tap. It demonstrates a good "data-ink" ratio and offers the user a way to zoom and navigate waveforms across time scales. It is easy to make complex soundscapes, but difficult to compose specific rhythms or pitches.

4. FUTURE WORK

Because sound file management is important to Twkyr's workflow, we will give Twkyr access to the host's iTunes library. Using Twkyr in the context of a networked ensemble will afford the opportunity to share state among instances of Twkyr, allowing each user to see other user's loop lengths and time scales in realtime. We will incorporate onset detection and tempo/pitch tracking to make Twkyr's "fret" visualizations and click-repair features more useful and allow "snapping" to an attack or a fixed metric position.

5. ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Robert W. Deutsch Foundation for funding this work and the players of the CREATE ensemble for workshopping this instrument. We would also like to thank UCSB's AlloSphere Research Group and the Experimental Visualization Laboratory for their support.

6. REFERENCES

[1] M. Alonso. Samplr. `http://samplr.net`.

[2] C. Carlson and G. Wang. Borderlands: An audiovisual interface for granular synthesis. In *Proc. NIME*, 2012.

[3] C. Roads. *Microsound*. MIT press, 2004.

[4] The Strange Agency. Curtis. `http://thestrangeagency.com/curtis-on-ipad`.

[5] E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, second edition, 2001.

[6] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.