The Stenophone: live coding on a chorded keyboard with continuous control

Jack Armitage
Queen Mary University of London
j.d.k.armitage@qmul.ac.uk

Andrew McPherson
Queen Mary University of London
a.mcpherson@qmul.ac.uk

ABSTRACT

Though laptop live coders are known to use other devices and instruments and play with other musicians, laptop live coding generally shares the common physical interface of the QWERTY keyboard. This project seeks to provide a means to explore alternatives to the QWERTY keyboard as a physical interface to laptop live coding. We present a live coding keyboard which is also a digital musical instrument, called the Stenophone. The Stenophone is an augmented stenotype or chorded keyboard, which permits continuous gestural control of keys and features an ergonomic design. These capabilities are exploited to enable the manipulation of algorithms and their parameterisation simultaneously.

1 Introduction

Early typewriters were inspired by the piano and built using piano parts, earning them the descriptor "literary piano" in Scientific American (1867), yet today the laptop as a musical instrument remains a controversial topic (Nilson 2007). Here we propose to remediate this history and blur the boundary further by replacing the QWERTY keyboard with a chorded typewriter keyboard. Whilst live coding is often regarded as belonging exclusively to a symbolic interaction paradigm, in this case we seek to combine it with gestures more readily associated with an embodied instrument (Magnusson 2009). Hardware laptop augmentations, such as key velocity sensing (Nash 2016) and hinge sensing (Meacham, Kannan, and Wang 2016), can provoke new approaches to programming, as in approximate programming where the programmer interacts with code "not just through the medium of text, but via any process that produces continuous numerical output" (Kiefer 2016). This project explores "gestural typing" by combining chorded typing and continuous control. Chording, stenotype, and continuous control are briefly reviewed here, and the work in progress instrument[1] is described.

1.1 Chorded keyboards and stenotype

Many musical instruments feature chording, where multiple finger digit inputs are combined to form a compound input. Chording is a feature of monophonic (trumpet, clarinet) and polyphonic (piano) instruments, and is often fundamental to performance. In early computer workstation research, chorded keyboards and mice were experimented with, but their steep learning curve was seen as an obstacle to their commercialisation (Engelbart and Lehtman 1988). However for many musicians chording is a common skill which can be recycled for live coding, to enable new exploration the long-standing problems of typing speed and direct expression (Collins et al. 2003).

The stenotype is a chorded keyboard featuring 22 keys developed for transcription and captioning via shorthand writing systems. Stenotype is superior to QWERTY and speech-to-text systems in speed and error rate; stenographers write at greater than 300 words per minute with no letter ordering mistakes. The ergonomic requirements of professional captioning mean modern stenotype keys feature continuous sensors which can be calibrated by the stenographer. The Open Steno Project[2] maintains open source steno hardware and software, making stenotype a solid basis for exploring chorded live coding.

[1] http://github.com/jarmitage/stenophone
[2] http://openstenoproject.org

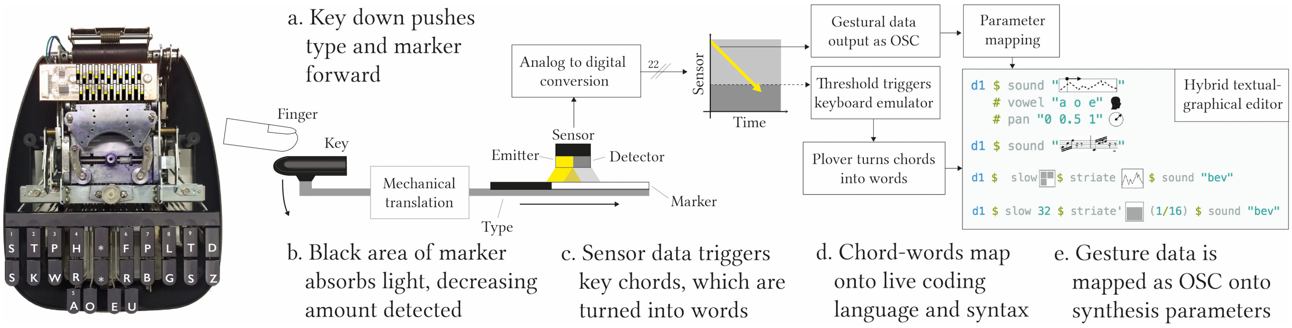

Figure 1: *Left: Top-down view of the Stenophone with top chassis removed, annotated with type letters and sensor/marker locations. Right: Diagram of system from mechanical input to live coding.*

1.2 Continuous control in digital musical instruments

Continuous control in digital musical instruments (DMIs) is increasingly common. Example instruments include TouchKeys, Seaboard, Linnstrument and Soundplane, and mobile devices with force-sensing displays (Lamb and Robertson 2011; McPherson 2012; McPherson 2015; Linn 2013; Jones et al. 2009). Typically the extra data are mapped to synthesis and modulation parameters to allow direct control over more complex sounds. Whilst continuous control data is usually distinguished from audio data and sampled at a lower rate, embedded platforms such as Bela make audio-rate sensing more feasible and suggest richer interaction possibilities (McPherson 2017).

Numerous efforts have proved the feasibility of keyboards that can sense surface touch, pressure, and hover gestures (Taylor et al. 2014; Dietz et al. 2009; Fallot-Burghardt et al. 2006). Proposed applications include integrating the mice into keyboards, kinetic typography, emotion recognition, biometric authentication, new gestural interactions, and programming (Lv and Wang 2006; Lv et al. 2008; Habib et al. 2009; Block, Gellersen, and Villar 2010). Bringing together continuous control and live coding through the computer keyboard is technically possible, but the pros and cons of the design space are currently uncharacterised.

2 Instrument design

2.1 Hardware

The first version of the Stenophone presented here is based on a used mechanical stenotype which was acquired at low cost to facilitate rapid prototyping. Professional stenotype machines can be costly, and whilst open source designs exist, the original mechanical keys are mounted on an array of metal levers and as such are ergonomically optimised and piano-like in their tactility. The stenotype machine itself was augmented to transmit key heights continuously via an Arduino-compatible Teensy USB microcontroller development board[3] over a serial port. Stenotype keys when pushed move the types towards the printer roll, and this motion was detected by attaching black and white markers to the types and measuring their optical reflectance (see Figure 1). A printed circuit board could be mounted on top of the types using existing screw holes, and the microcontroller was installed in the paper tray to preserve the original design as much as possible.

2.2 Software

Processing[4] was used to calibrate the sensor data and output keypresses when the keys cross a threshold, and key heights continuously as OSC data. The keypresses are interpreted by open source stenotype software Plover.[5] Plover reads keyboard input and groups keypresses into chords and translates chords into words via a customisable lookup dictionary. It displays the latest shorthand chord entries in a window as would be seen on an original stenotype printer roll. Users can build custom shorthand dictionaries according to their specialism, and in this case shorthand chords were optimised for live coding in Tidal (McLean 2014). As an example, one could write `d1 $ every 2 (fast 2) $ sound "drum"` using the chords shown in Table 1, which would be 11 chords instead of 36 individually typed characters, with zero spelling errors guaranteed (Plover also facilitates rapid chord error correction).

[3] http://pjrc.com/teensy
[4] http://processing.org
[5] http://openstenoproject.org/plover

Table 1: Example phonetic-mnemonic shorthand chords and Tidal translations in a Plover dictionary.

Shorthand chord	S*D	TK-PL	EFR	NUMBER BAR + T	PREPB	TPAFT	SOUPBD	KW-GS	TKRUPL
Tidal translation	d1	$	every	2	(	fast	sound	“	drum

2.3 Mapping approaches

As well as augmenting live coding through chorded input, the gestural data from the 22 keys can also be mapped in many different ways to provide further augmentation. Exploration of this new creative space is in progress and will require further development and practise in order to evaluate. It is possible at this stage to briefly outline the conceptual approaches and driving questions being considered when using the Stenophone with existing live coding systems.

The Stenophone's gestural data can be input into the text editor, the synthesis engine, or both. Gestures can manipulate text, number patterns and structure in the text editor itself, or they can map onto synthesis parameters without being converted to text. In the latter case, gestures can be recorded during chord input and used to parameterise wavetables, envelopes and other effects. The concepts of modal and modeless interface are potentially useful, to describe whether the same input produces different outputs (as in the caps lock key) or not. A modal Stenophone interface might allow switching between modes for words, piano notes and parameter curves, whereas a modeless interface might rely on for example the cursor position to contextually determine gestural output.

3 Discussion

Laptop live coding can be a stressful and frustrating experience for performers and audiences. For the performer stress may arise due to typing or software errors, or due to making too few musical changes, too slowly, at low levels of musicality. The traditional approach to live coding has been to embrace these imperfections, or claim that the compromises are necessary to access pure abstraction (Nilson 2007). An alternative idea is to make programming more tactile, gestural and ultimately more tacit, whilst retaining the basic form of the laptop. The Stenophone has demonstrated such an approach, which could transfer some of the complexity of live coding from the brain to the hands, for those interested in doing so.

The hardware and software design of the initial prototype of the Stenophone was deliberately limited to lower costs and facilitate rapid development. Future versions could add more dimensions of gesture to the keys (such as touch and hover) and more resolution by sampling at audio rate. New live coding software could be developed around gestural, chorded keyboards, which are more natively able to support gestural data. The language Tidal was used for prototyping, but it is possible that the Stenophone might be better suited to a stack-based, concatenative language[6] where input is sequential. As in the origins of live coding, these ideas will need to be trialled through extensive practise and refinement, and subject to audience feedback.

4 Conclusion

The Stenophone is a viable basis for augmenting laptop live coding, where programming and musical gesture are integrated in a single musical instrument. The Stenophone offers advantages to live coders due to being based on a stenotype system, and opens up new possibilities for live coding performance and research. The Stenophone demonstrates the potential for programming as a physical activity to diversify beyond the QWERTY keyboard. This illustrates an opportunity for further research into how the activity of programming can be changed to suit the physical skills and cultural practices of performing musicians.

4.1 Acknowledgments

This research is supported by EPSRC under grants EP/G03723X/1 and EP/L01632X/1 (Centre for Doctoral Training in Media and Arts Technology).

[6]For example; Møllersen 2017. *Ait: A Concatenative Language for Creative Programming*, http://functional-art.org/2017/ait

References

Block, Florian, Hans Gellersen, and Nicolas Villar. 2010. "Touch-Display Keyboards: Transforming Keyboards into Interactive Surfaces." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1145–54. ACM.

Collins, Nick, Alex McLean, Julian Rohrhuber, and Adrian Ward. 2003. "Live Coding in Laptop Performance." *Organised Sound* 8 (3): 321–30.

Dietz, Paul H., Benjamin Eidelson, Jonathan Westhues, and Steven Bathiche. 2009. "A Practical Pressure Sensitive Computer Keyboard." In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, 55–58. ACM.

Engelbart, Douglas, and Harvey Lehtman. 1988. "Working Together." *Byte* 13 (13): 245–52.

Fallot-Burghardt, Wolfgang, Morten Fjeld, C. Speirs, S. Ziegenspeck, Helmut Krueger, and Thomas Läubli. 2006. "Touch&Type: A Novel Pointing Device for Notebook Computers." In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles*, 465–68. ACM.

Habib, Iman, Niklas Berggren, Erik Rehn, Gustav Josefsson, Andreas Kunz, and Morten Fjeld. 2009. "Dgts: Integrated Typing and Pointing." *Human-Computer Interaction–INTERACT 2009*. Springer, 232–35.

Jones, Randy, Peter F. Driessen, W. Andrew Schloss, and George Tzanetakis. 2009. "A Force-Sensitive Surface for Intimate Control." In *Proc. NIME 2009*, 236–41.

Kiefer, Chris. 2016. "ApProgXimate Audio: A Distributed Interactive Experiment in Sound Art and Live Coding." *International Journal of Performance Arts and Digital Media* 12 (2): 195–200.

Lamb, Roland, and Andrew Robertson. 2011. "Seaboard: A New Piano Keyboard-Related Interface Combining Discrete and Continuous Control." In *NIME*, 503–6.

Linn, Roger. 2013. "LinnStrument and Other New Expressive Musical Controllers." *The Journal of the Acoustical Society of America* 134 (5): 4053–3.

Lv, Hai-Rong, and Wen-Yuan Wang. 2006. "Biologic Verification Based on Pressure Sensor Keyboards and Classifier Fusion Techniques." *Consumer Electronics, IEEE Transactions on* 52 (3): 1057–63.

Lv, Hai-Rong, Zhong-Lin Lin, Wen-Jun Yin, and Jin Dong. 2008. "Emotion Recognition Based on Pressure Sensor Keyboards." In *Multimedia and Expo, 2008 IEEE International Conference on*, 1089–92. IEEE.

Magnusson, Thor. 2009. "Of Epistemic Tools: Musical Instruments as Cognitive Extensions." *Organised Sound* 14 (02): 168–76.

McLean, Alex. 2014. "Making Programming Languages to Dance to: Live Coding with Tidal." In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design*, 63–70. ACM.

McPherson, Andrew. 2012. "TouchKeys: Capacitive Multi-Touch Sensing on a Physical Keyboard." In *NIME*.

McPherson, Andrew. 2015. "Buttons, Handles, and Keys: Advances in Continuous-Control Keyboard Instruments." *Computer Music Journal*.

———. 2017. "Bela: An Embedded Platform for Low-Latency Feedback Control of Sound." *The Journal of the Acoustical Society of America* 141 (5). ASA: 3618–8.

Meacham, Aidan, Sanjay Kannan, and Ge Wang. 2016. "The Laptop Accordion." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 16:236–40. 2220-4806. Brisbane, Australia.

Nash, Chris. 2016. "The 'E' in QWERTY: Musical Expression with Old Computer Interfaces." In *Proc. Nime 2016*, 16:224–29. 2220-4806. Brisbane, Australia: Queensland Conservatorium Griffith University.

Nilson, Click. 2007. "Live Coding Practice." In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 112–17. ACM.

Scientific American. 1867. "Type Writing Machine." *Scientific American*, July.

Taylor, Stuart, Cem Keskin, Otmar Hilliges, Shahram Izadi, and John Helmes. 2014. "Type-Hover-Swipe in 96 Bytes: A Motion Sensing Mechanical Keyboard." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1695–1704. ACM.