DACTYLIZE: AUTOMATICALLY COLLECTING PIANO FINGERING DATA FROM PERFORMANCE

David A. Randolph and Barbara Di Eugenio
University of Illinois at Chicago
Department of Computer Science
{drando2,bdieugen}@uic.edu

ABSTRACT

A prototype system, dubbed "Dactylize," for collecting fingering data automatically from actual piano performances is described. The solution promises to be an economical and accurate tool for developing corpora related to piano fingering. Evaluation of an early prototype suggests accuracy over 99% at rates up to 12.5 notes per second.

1. OVERVIEW

Extensive corpora are essential for developing both cognitive and computational models. To date, published models developed for the piano fingering problem have relied on small amounts of manually created data for training and/or evaluation [7, 8, 10, 12, 16] or on humans to validate system output *ad hoc* [1, 2]. Other models' authors [5, 15], absent a gold-standard corpus, make no specific performance claims. To our knowledge, no significant publicly available machine-readable datasets exist for this domain. The system described here will help address this deficit.

Compared to previously suggested computer vision [4, 9, 11, 14] and vibration sensing [6] systems for piano fingering detection, our approach is straightforward. We overlay each key of a MIDI-enabled keyboard with foil tape and wire each foil area as digital input to a micro-controller circuit. Similarly, we apply conductive fabric tape to the area on each finger that contacts the keys and wire these contacts as digital input. The circuit implements a patch bay and detects when a particular finger is in contact with a particular key. Essentially, the key wire, key foil, finger tape, and finger wire collectively form a patch cord. Couple this with a synchronized flow of MIDI events, and we are able to determine accurately which finger played each note.

The cost to develop the system, as of this writing, is approximately $300 in addition to the cost of the underlying digital piano. A photo album chronicling the initial system build is at https://goo.gl/photos/nrru4XtyrzUwzhcS6.

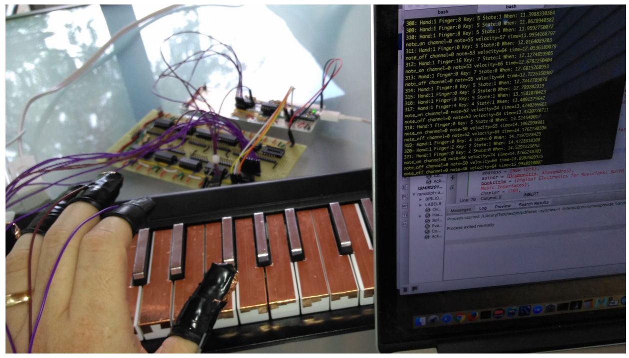

Figure 1. Monitor output from demonstration system.

2. HARDWARE

2.1 Digital Piano

Any MIDI-enabled keyboard may be used to build a Dactylize system. While the modifications suggested here are reversible, the process will likely exact a toll beyond normal wear and tear. While our portable demonstration system uses an inexpensive Korg microKEY, for serious data collection, the best obtainable full-sized, graded, hammer-weighted digital piano should be used. Faithful capture in MIDI of nuances in performance may prove essential in future analysis. We have modified a Casio Privia PX-130 that is several years old to serve in our production system. The Privia systems are well regarded inexpensive instruments. The current $500 PX-160 is also recommended.

2.2 Key Wiring

The white keys of our production system have each been covered in aluminum foil tape (inexpensive tape intended for ductwork). The overlay for each key has been cut to fit, with care taken to ensure that no adjoining overlays touch each other. For contrast, half-inch strips from one-inch copper foil tape have been used for the black keys.

Using small pieces of copper foil tape, the female ends of 0.1-inch jumper wires have been attached to the distal surfaces of each key's foil overlay. These wires are collected in groups of eight using zip ties and secured to the piano housing using self-adhesive zip tie mounts. The other (male) ends of the jumper wires are eventually joined to 2x8-pin housings and connected to ribbon cables that lead to the micro-controller circuit.

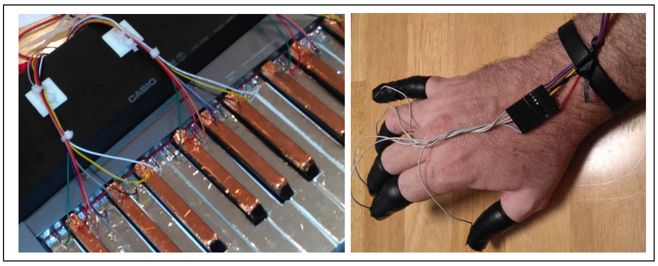

Figure 2. Wired key foil overlays on production system (left) and finger attachments (right).

2.3 Finger Wiring

Each finger attachment is constructed by affixing an octagonal piece—with a short rectangular extension—of conductive (nickel, copper, and cobalt) ripstop fabric tape to the tip of a latex finger cot. An additional section of tape is added to cover the outside of the thumb, an area commonly contacting the keys. Stripped 6-inch stranded wire is attached to each extension with copper foil tape and further secured with electrical tape. Wires are collected in a 1x8-pin Dupont-style housing and mated to a ribbon cable connecting to the circuit. To reduce movement of the hand assembly, wires are gathered and attached to a silicone bracelet at the player's wrist with a twist tie.

2.4 Micro-Controller Circuit

Closely following Drymonitis [3], we have implemented a 16x79 patch-bay matrix circuit comprising an input shift register circuit (with two daisy-chained 74HC165 integrated circuit chips to cover 10 fingers), an output shift register circuit (ten 74HC595 chips to cover 79 keys), and an Osoyoo Micro Pro (Arduino clone) micro-controller to drive the monitoring activity. See Figure 4 for details.

3. SOFTWARE

The micro-controller code, again based largely on Drymonitis's implementation, polls the circuit continuously for changes in the connection state between key pins and finger pins. It sends its output serially via USB to a personal computer, which is also connected via USB to the digital piano. A monitoring script, written in Python and leveraging the *serial* and *mido* modules, captures and time stamps (with microsecond granularity) the serial and MIDI output and saves these data to separate files. Finally, when a performance is complete, a Perl script is executed to process the files produced by the monitor, inferring a fingering for each note and outputting a fingered score in a machine-readable format (as MIDI and abcDF [13]).

The algorithm for determining these fingerings is currently quite simple. It loops through the MIDI notes and assigns the "striking" finger as the last one to contact the note prior to its onset time. [1] The "releasing" finger is the last finger seen to touch the key prior to its note's ending time stamp, or, if no such finger contact is detected, the striking and releasing fingers are assumed to be the same.

All code is open source and is maintained on GitHub at https://github.com/dvdrndlph/dactylize.

[1] If a release event is the last one detected for a key prior to the note onset, the algorithm will accept the first contact event prior to the note's ending time. Such an event sequence is possible given the circuit latency.

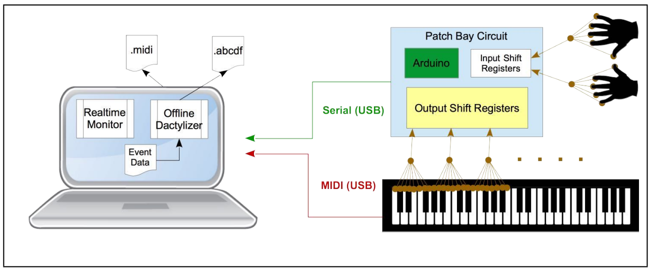

Figure 3. Dactylize architecture. *Credits: Piano by Juan Pablo Bravo and hand by Dmitry Baranovskiy, both from the Noun Project.*

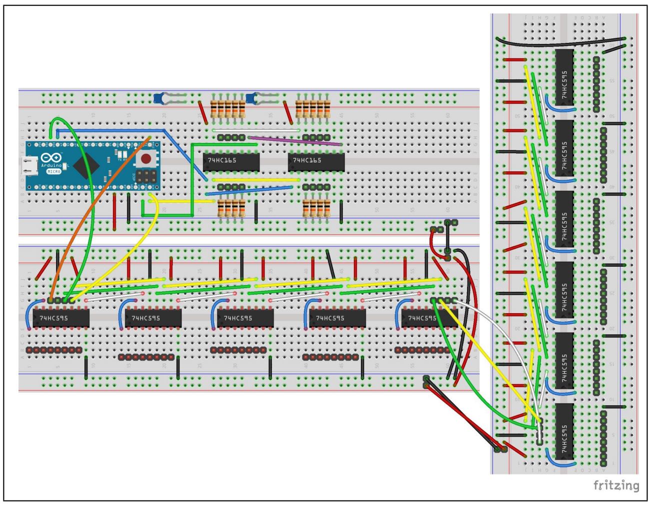

Figure 4. Micro-controller circuit diagram.

4. EVALUATION

As a preliminary test of system performance, the first author (a non-pianist) performed a simple passage on the white keys only of our smaller demonstration system (with key width of 0.75 inches, compared to standard 0.94 inch key width), ascending from C to G and returning to C. The passage was repeated three times for a total of 25 notes per trial. Tempo was increased from 4.2 to 12.5 notes per second across 19 trials. We discarded two trials in which incorrect notes were struck. The system correctly identified the finger for 421 of 425 notes, implying accuracy over 99.0%. Furthermore, the system misidentified no notes.

Still, usability challenges remain. Alternative materials are being investigated to make the finger assemblies more robust and less unobtrusive. The micro-controller circuit is being debugged to extend support to all 88 keys.

5. ACKNOWLEDGMENTS

Thanks to Alexander Demos for suggesting this general approach and to David Meyer and Alex Radosavljevic. This work has been supported by a Provost's Award for Graduate Research from the University of Illinois at Chicago.

6. REFERENCES

[1] Matteo Balliauw, Dorien Herremans, Daniel Palhazi Cuervo, and Sörensen. Generating fingerings for polyphonic piano music with a tabu search algorithm. In *Proceedings of the International Conference on Mathematics and Computation in Music*, pages 149–160, London, 2015. Springer.

[2] Roberto De Prisco, Gianluca Zaccagnino, and Rocco Zaccagnino. A differential evolution algorithm assisted by ANFIS for music fingering. In Leszek Rutkowski, Marcin Korytkowski, Rafał Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Swarm and Evolutionary Computation*, volume 7269 of *Lecture Notes in Computer Science*, pages 48–56, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[3] Alexandros Drymonitis. A Patch-Bay Matrix Synthesizer. In *Digital Electronics for Musicians: Build Intuitive Electronic and Electroacoustic Music Interfaces*, chapter 10, pages 417–480. Apress, New York, 2015.

[4] Dimitry Gorodnichy and A. Yogeswaran. Detection and tracking of pianist hands and fingers. In *Proceedings of the Third Canadian Conference on Computer and Robot Vision*, Quebec, 2006. National Research Council Canada.

[5] Melanie Hart, Robert Bosch, and Elbert Tsai. Finding Optimal Piano Fingerings. *The UMAP Journal*, 21(2):167–177, 2000.

[6] Chih-pin Hsiao, Richard Li, Xinyan Yan, and Ellen Yiluen Do. Tactile Teacher: Sensing finger Tapping in Piano Playing. In *Proceedings of the 9th Conference on Tangible, Embedded and Embodied Interaction*, pages 257–260, Stanford, California, USA, 2015.

[7] J. Pieter Jacobs. Refinements to the ergonomic model for keyboard fingering of Parncutt, Sloboda, Clarke, Raekallio, and Desain. *Music Perception*, 18(4):505–511, 2001.

[8] Alia Al Kasimi, Eric Nichols, and Christopher Raphael. A simple algorithm for automatic generation of polyphonic piano fingerings. In Simon Dixon, David Bainbridge, and Rainer Typke, editors, *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 355–356, Vienna, 2007.

[9] Jennifer Macritchie and Nicholas J. Bailey. Efficient tracking of pianists' finger movements. *Journal of New Music Research*, 42(1):79–95, 2013.

[10] Eita Nakamura, Nobutaka Ono, and Shigeki Sagayama. Merged-output HMM for piano fingering of both hands. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 531–536, Taipei, Taiwan, 2014.

[11] Akiya Oka and Manabu Hashimoto. Marker-less piano fingering recognition using sequential depth images. In *Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, volume 4, pages 1–4, Incheon, South Korea, 2013. IEEE.

[12] Richard Parncutt, John A. Sloboda, Eric F. Clarke, Matti Raekallio, and Peter Desain. An ergonomic model of keyboard fingering for melodic fragments. *Music Perception*, 14(4):341–382, 1997.

[13] David A. Randolph and Barbara Di Eugenio. Easy as abcDE: Piano fingering transcription online. In *Extended Abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*, 2016 (in press).

[14] Yoshinari Takegawa and Masahiko Tsukamoto. Design and implementation of a piano practice support system using a real-time fingering recognition technique. In *Proceedings of the International Computer Music Conference*, pages 387–394, University of Huddersfield, UK, 2011.

[15] Alexandre B. Viana, Jose H. F. Cavalcanti, and Pablo Javier Alsina. Intelligent system for piano fingering learning aid. In *Proceedings of the Fifth International Conference on Control, Automation, Robotics & Vision*, Singapore, 1998.

[16] Yuichiro Yonebayashi, Hirokazu Kameoka, and Shigeki Sagayama. Automatic decision of piano fingering based on a hidden markov models. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 2915–2921, 2007.