KeyFlow: Acoustic Motion Sensing for Cursor Control on Any Keyboard

Yujia Liu*
Tsinghua University
Beijing, China

Qihang Shan*
Virginia Tech
Blacksburg, VA, USA

Zhihao Yao
Tsinghua University
Beijing, China

Qiuyu Lu†
University of California, Berkeley
Berkeley, CA, USA

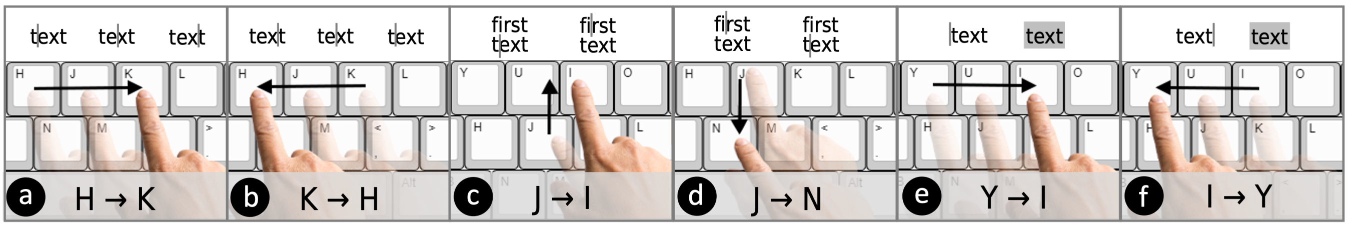

Figure 1: KeyFlow keyboard glide gestures. KeyFlow integrates mouse functionality into the keyboard, allowing users to glide fingers to move the cursor (a-d) and select words (e-f) without pressing keys.

ABSTRACT

Despite typing being a critical operation in the digital age, users still need to frequently switch between the mouse and keyboard while typing. We introduce KeyFlow, a tool that integrates mouse functionality into the keyboard through machine learning, allowing users to glide their fingers across the keyboard surface to move the cursor. The whole process does not press the keys down to differentiate from normal typing and avoid false touches. KeyFlow uses any computer-built-in microphones to capture the acoustic features of these gliding gestures, requiring no specialized equipment and can be set up and tested independently within 5 minutes. Our user research indicates that, compared to traditional keyboard and mouse methods, this system reduces hand movement distance by 78.3%, making the typing experience more focused.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Keyboard, Typing Experience, Gesture Recognition, Passive Acoustic Sensing, Machine Learning

*Both authors contributed equally to this research.
†The corresponding author: qiuyulu@berkeley.edu

UIST Adjunct '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0718-6/24/10.
https://doi.org/10.1145/3672539.3686348

ACM Reference Format:
Yujia Liu, Qihang Shan, Zhihao Yao, and Qiuyu Lu. 2024. KeyFlow: Acoustic Motion Sensing for Cursor Control on Any Keyboard. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3672539.3686348

1 INTRODUCTION

As digital communication continues to grow, proficient typing is increasingly important. Therefore, how to improve typing efficiency and experience has become the focus of researchers. Innovations in this area include autocorrection [11, 16], virtual soft keyboards [4, 9, 15], keyboard layouts [1], and auxiliary devices [10, 12]. A significant challenge in maintaining typing fluency is moving the cursor while typing, as switching between the keyboard and the mouse takes an average of 0.36 seconds each time [3], and using the arrow keys takes 0.21 seconds each time [3], both of which disrupt typing flow. Some researchers [5] have attempted motion sensing through keyboard presses, but this approach suffers from false touches. Other researchers have used additional sensors [10, 12] or desktop cameras [13, 14] to capture motion. However, these methods require extra hardware, which is inefficient and cumbersome to set up and maintain.

We introduce KeyFlow (Figure 1), an innovative software solution that uses acoustic motion sensing for cursor control on any keyboard, eliminating the need for additional hardware. KeyFlow enhances typing fluency by allowing users to keep their hands in place and move the text cursor with simple finger slides on the keyboard. The acoustic features can be used to distinguish different motions because each key's unique mechanical structure produces distinctive sound changes when a finger slides over it [10]. Inspired by this insight, KeyFlow employs a passive acoustic sensing method

[6–8], which requires minimal hardware, any computer's built-in or low-cost microphone, along with correction algorithms, to classify the sounds of sliding across the keyboard. This paper utilizes the widely-used SVM as the classification tool and MFCC as the training feature, comprising 64 one-dimensional data points, requiring minimal training data, running at high speed, and with low computational requirements. We provide user-friendly instructions and training tools, allowing any novice user to configure the system easily within 5 minutes. During use, the user only needs to slide their finger lightly over the keyboard without pressing any keys, to move the cursor, select words, and scroll. This makes the typing experience more seamless and effortless, reducing hand movement distance by 78.3% compared to traditional mouse-keyboard methods. Additionally, KeyFlow supports the importation of presets and the creation of personalized gestures, further enhancing its utility.

2 KEYFLOW

KeyFlow defines 3 functions(move, select, scroll) and 4 motions(left, right, up, down). In our examples, we preset 6 easily accessible keyboard keys - YUIHJK (Fig 1), and users can also customize their own combinations.

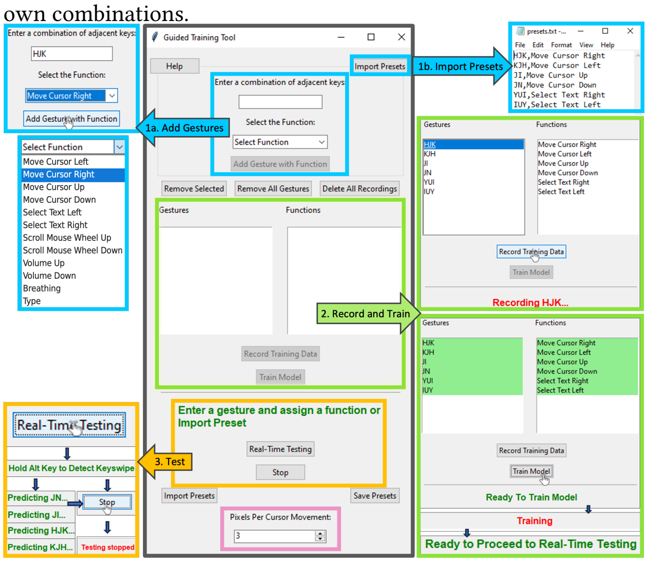

Figure 2: User Interface and Workflow. (1) Setup: Users can either (1a) customize key combinations and functions or (1b) import presets. (2) Record and Train: Record gestures, then train the model. Status updates are shown in red for recording and green upon completion. (3) Testing: Real-time testing begins by holding the "Ctrl"; it can be paused or resumed anytime. (4) Adjustment: Customize how many characters the text cursor moves at a time.

Sound Capture. 3 microphones capture sounds at a 44100 Hz sampling rate, with each sample lasting 0.0232 seconds (1024 data points). Each gesture requires 6 seconds of recorded data.

Machine Learning Setup. We use MFCCs as features due to their high accuracy in key tap classification. We evaluated SVM, MLPNN, KNN, and Tree models. SVM was chosen for its better performance, lower hardware requirements, and shorter training time (table 1). We also use a "lag and vote" system that outputs the most frequent prediction from the latest three, reducing lag to 0.0232 seconds.

The software part was written entirely using Python and is available in GitHub repository. Moving cursor is enabled via the pyautogui Python library.

Model	TPR	ACC	Training Time
SVM	0.76	0.93	0.015s
MLPNN	0.75	0.92	0.277s
KNN	0.67	0.89	0.026s
Tree	0.37	0.79	0.657s

Table 1: Comparison of Machine Learning Models

3 EVALUATION

This program was assessed in 2 aspects: machine learning model accuracy with different hardware, and performance evaluation from 12 users.

Technical Review. The model was evaluated with 6 seconds of training and 9 seconds of testing samples, achieving a True Positive Rate (TPR) of 0.797 and an Accuracy (ACC) of 0.933. The 6-feature model completes its training in 0.696 seconds, indicating fast processing with low computational demands. Moreover, accuracy tests across 6 setups involved a range of keyboards, including Alienware, Logitech, Surface, and MacBook, as well as microphones like the Razer Kraken and built-in options.

User Experience. Twelve users evaluated KeyFlow on assessing training time, model accuracy, and performance in typing and deleting tasks compared to conventional methods. After using the tool, they completed a survey incorporating the System Usability Scale [2] and custom questions on lag and efficiency. The average scores (out of 5) were: System Preference (3.67), Simplicity (4.01), Consistency (3.88), Accessibility (4.31), Timeliness (4.75), System Efficiency (4.25), and Subjective Accuracy (4.72).

The findings highlight a preference for this tool over traditional keyboard-mouse setups due to its simplicity, consistency, and user-friendliness. Setup time averaged 2 minutes and 59 seconds, suitable even for those with minimal machine-learning expertise. The model demonstrated accuracy between 90%-95%. In performance tests, deleting words was 15.9% faster and deleting letters 14.0% quicker than with conventional methods. Most notably, the tool reduced necessary arm movement for text correction by 78.3%, significantly enhancing usability and efficiency.

4 DISCUSSION AND FUTURE WORK

KeyFlow integrates mouse functionality into keyboards, reducing hand movement by 78.3%. However, it relies on built-in microphones, which are not standard in most keyboards. Future work should explore cost-effective microphones and optimal layouts.

Performance drops in noisy environments, with True Positive Rate falling from 0.797 to 0.64 and Accuracy from 0.93 to 0.88. Incorporating noise cancellation algorithms could address this.

Comparative studies with software and hardware are necessary to validate efficiency. A user study should evaluate the impact of accuracy on typing speed and experience.

Refining the user interface to eliminate hotkeys and simplify customization will enhance usability. Addressing these areas will make KeyFlow more practical and widely applicable.

REFERENCES

[1] Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2010. Quasi-Qwerty Soft Keyboard Optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 283–286. https://doi.org/10.1145/1753326.1753367

[2] John Brooke. 1996. Sus: a "quick and dirty'usability. *Usability evaluation in industry* 189, 3 (1996), 189–194.

[3] Stuart K Card, William K English, and Betty J Burr. 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics* 21, 8 (1978), 601–613.

[4] Wenqiang Chen, Yanming Lian, Lu Wang, Rukhsana Ruby, Wen Hu, and Kaishun Wu. 2017. Virtual Keyboard for Wearable Wristbands. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (Delft, Netherlands) *(SenSys '17)*. Association for Computing Machinery, New York, NY, USA, Article 44, 2 pages. https://doi.org/10.1145/3131672.3136984

[5] Wolfgang Fallot-Burghardt, 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*. 465–468.

[6] Mayank Goel, Brendan Lee, Md. Tanvir Islam Aumi, Shwetak Patel, Gaetano Borriello, Stacie Hibino, and Bo Begole. 2014. SurfaceLink: Using Inertial and Acoustic Sensing to Enable Multi-Device Interaction on a Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1387–1396. https://doi.org/10.1145/2556288.2557120

[7] Teng Han, Khalad Hasan, Keisuke Nakamura, Randy Gomez, and Pourang Irani. 2017. SoundCraft: Enabling Spatial Interactions on Smartwatches Using Hand Generated Acoustics. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 579–591. https://doi.org/10.1145/3126594.3126612

[8] Chris Harrison and Scott E. Hudson. 2008. Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile Finger Input Surfaces. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 205–208. https://doi.org/10.1145/1449715.1449747

[9] Jinki Jung, Jinwoo Jeon, Hyeopwoo Lee, Kichan Kwon, Jamal Zemerly, and Hyun Seung Yang. 2014. Augmented Keyboard: A Virtual Keyboard Interface for Smart Glasses. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Shenzhen, China) *(VRCAI '14)*. Association for Computing Machinery, New York, NY, USA, 159–164. https://doi.org/10.1145/2670473.2670482

[10] Ülkü Meteriz Yýldýran, Necip Fazýl Yýldýran, and David Mohaisen. 2022. AcousticType: Smartwatch-Enabled Cross-Device Text Entry Method Using Keyboard Acoustics. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 352, 7 pages. https://doi.org/10.1145/3491101.3519691

[11] OpenAI. 2022. *AI-powered language model developed by OpenAI.* OpenAI. https://chat.openai.com/

[12] Stuart Taylor, 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* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1695–1704. https://doi.org/10.1145/2556288.2557030

[13] Robert Wang, Sylvain Paris, and Jovan Popović. 2011. 6D hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 549–558.

[14] Andrew D Wilson. 2006. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 255–258.

[15] Naoki Yanagihara and Buntarou Shizuki. 2018. Cubic Keyboard for Virtual Reality. In *Proceedings of the 2018 ACM Symposium on Spatial User Interaction* (Berlin, Germany) *(SUI '18)*. Association for Computing Machinery, New York, NY, USA, 170. https://doi.org/10.1145/3267782.3274687

[16] Mingrui Ray Zhang, He Wen, and Jacob O. Wobbrock. 2019. Type, Then Correct: Intelligent Text Correction Techniques for Mobile Text Entry Using Neural Networks. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 843–855. https://doi.org/10.1145/3332165.3347924