Augmenting Tablet Typing Experience by Integrating Key-Press Finger Contact Types as Input

Yantao Liu
Department of Computer Science and Technology
Shantou University
Shantou, China
23ytliu@stu.edu.cn

Dongmin Xiao
Department of Computer Science and Technology
Shantou University
Shantou, China
23dmxiao@stu.edu.cn

Taizhou Chen*
Department of Computer Science and Technology
Shantou University
Shantou, China
tzchen@stu.edu.cn

Kening Zhu
School of Creative Media
City University of Hong Kong
HongKong, China
keninzhu@cityu.edu.hk

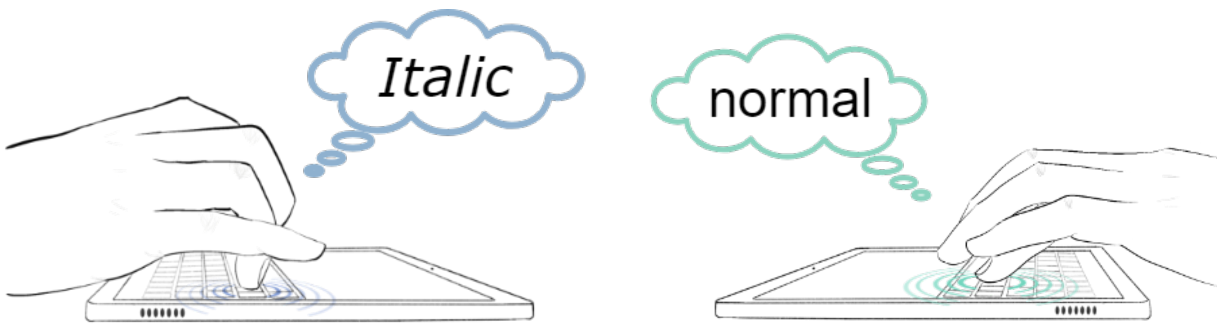

Figure 1: Left: Finger tip contact (FT). Right: Finger pad contact (FP).

Abstract

Touchscreen typing on tablet has become popular in modern digital routine, calling for investigation of more expressive input method on touchscreen keyboard. In this paper, we propose a novel approach to augment touchscreen typing experience by integrating key-press finger posture recognition to extend the input space of a standard touchscreen keyboard. Our system distinguish between finger tip contact(FP) and finger pad contact(FP) through acoustic sensing, enabling seamless switching between normal and functional input. We evaluate the performance of our system through offline and online experiments, where we show that our system achieves an offline key-wise recognition accuracy of up to 96.3%. The online experiment shows a real-time recognition accuracy of 94% and 88% in quiet and noisy environments, respectively. We further conducted a usability study on text formatting task, which shows that our method significantly outperform the baseline method in terms of input speed and functionality.

*Corresponding author.

CHI EA '25, Yokohama, Japan

ACM ISBN 979-8-4007-1395-8/25/04
https://doi.org/10.1145/3706599.3720116

CCS Concepts

• **Human-centered computing** → **Text input**; *Gestural input*; Empirical studies in HCI.

Keywords

Touchscreen Interaction, Text Input, Acoustic Sensing

ACM Reference Format:
Yantao Liu, Dongmin Xiao, Taizhou Chen, and Kening Zhu. 2025. Augmenting Tablet Typing Experience by Integrating Key-Press Finger Contact Types as Input. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3706599.3720116

1 Introduction

Touchscreen typing has become an essential practice in modern digital routines, given the proliferation of smartphones, tablets, and other touch-enabled devices in our daily lives. Specifically, more and more people prefer to use tablets for text entry tasks such as taking notes, writing emails, and office work[19], as the tablet offers a portable and intuitive interface, allowing users to perform tasks efficiently. Although text composition remains the dominant task for text entry, various text editing tasks, such as cut, copy, paste, and text formatting[30, 35], are also essential, calling for investigation of more expressive interaction on the keyboard. Compared to desktop devices with physical keyboards that provide extensive support for shortcut-based command input (e.g. hotkeys[26]), touchscreen keyboards do not offer such a unified shortcut feature[6]. Although touchscreens are capable of providing rich and intuitive input for general interactions, the transition between touch-based navigation and typing[8, 18] often disrupts the user experience, making touchscreen typing less efficient for complex text editing tasks[19].

To alleviate this issue, several researches explored designing on-keyboard gestures for tablet[4, 7, 21, 22] to extend the functionality of touchscreen keyboards. While on-keyboard gestures offer an intuitive way to enhance the typing experience, performing gesture also disrupts the typing experience. On the other hand, numerous researches have dedicated themselves to exploring enhancing touchscreen-based input modality by detecting on-screen finger posture through various approaches such as raw capacitive images[1, 14, 24, 36], acoustic[12, 20], and external wearables[10, 16, 29]. However, there still remains a notable gap in research on improving the typing experience, especially on tablets.

In this paper, we present a novel typing technique that seamlessly integrates functional inputs for typing tasks on tablet touchscreen keyboards. Our system extends the input space of every single key on a tablet touchscreen keyboard by distinguishing two types of finger contact, namely finger tip contact (FT) and finger pad contact (FP). By seamlessly switching between two finger contact types during typing, users are allowed to perform additional actions, such as text formatting commands or text editing shortcuts directly within the typing interface. We implement an augmented touchscreen keyboard that enables real-time finger contact types recognition through acoustic sensing. Our system can be deployed on any off-the-shelf tablet without requiring any external hardware or manipulating the system kernel. The offline experiment shows that our system achieved an average general accuracy of 93.5% and an average user-dependent accuracy of up to 96.3%. Additionally, an online between-subject user study indicates that our system performs robustly in both quiet and noisy environments, with key-wise accuracies of 88% and 94%, respectively. We further conduct a system usability study on a text formatting task using our method. The result shows that users complete the task significantly faster with our system compared to the traditional one.

2 Related Works

Our work is largely inspired by existing works on touch posture sensing on touchscreens and augmented touchscreen keyboard for typing.

2.1 Touch Posture Sensing on Touchscreen

Touchscreen-based interaction dominates the input channel on most modern mobile devices, including smartphones, tablets, and smartwatches. Numerous research works have dedicated themselves to extending the input space of a touchscreen by detecting touch posture. TapSense[12] detects finger impact parts including tip, pad, nail, and knuckle on a touchscreen through acoustic sensing. They collect data and evaluate their concept on a proof-of-concept device with a back-mounted stethoscope. In addition to the acoustic-based method, leveraging the raw capacitive image of finger-touching patterns to retrieve the hand posture is a common approach in the community. Xiao et al.[36] proposed a pioneering work to estimate the 3D finger angle on both smartphone and smartwatch utilizing raw capacitive image. They built a proof-of-concept device by modifying the system kernel to obtain raw capacitive image data for estimating 3D finger angle, which inspire many following works on exploring raw capacitive image data to enrich on-screen interaction[13, 14]. Raw capacitive images have also been utilized to detect contact fingers on the touchscreen of devices such as smartphone[23], tablet[17], and smartwatch[9]. Researches also leveraged external hardware to sense touch posture, including finger-worn optical sensor[10, 11], forearm-worn electromyography measuring devices[2], and ring-form devices[16, 29].

Although the aforementioned works offer solutions for detecting on-screen touch posture in different contexts, there are still facing limitations. For example, raw capacitive data can only be obtained by manipulating the system kernel[13, 14, 36], which offers extra burden for the deployment on commercial devices. For literature that requires external hardware would restrict the system flexibility in a mobile context. We extend Harrison et al.'s idea[12] by implementing and evaluating a real-time touch posture sensing system to facilitate typing experiences. Our system offers a fast and robust real-time touch posture sensing approach for touchscreen keyboard without requiring any external hardware or kernel manipulation that can be easily deployed on any off-the-shelf device.

2.2 Augmented Touchscreen Keyboard for Typing

Various approaches have been proposed to facilitate the typing experience by augmenting the touchscreen keyboard. Researches have explored improving typing efficiency by adaptively optimizing the touchscreen layout[15, 37]. Numerous researches also dedicated themselves on exploring gesture-based input to facilitate text entry efficiency[32, 34] and text editing experiences[21, 22, 31]. While gesture keyboards[4] are increasingly popular as it shows an advantage to improve the touchscreen keyboard typing speed, they are suffering from the gesture ambiguity problem that would increase the typing error rate[3]. Smith et al.[34] investigated a keyboard layout optimization method to reduce the error rate of a gestural touchscreen QWERTY keyboard by 52%. Jochen Rick[32] evaluated through 22 virtual keyboard layouts using a Fitts's Law-based model to explore an optimal keyboard layout for gesture typing. GeShort[31] proposed gesture-based input to facilitate command-based text editing and formatting tasks about 11% and 22% faster respectively. Per Ola Kristensson and Shumin Zhai[22] introduced a series of stylus-based stroke gestures to enable fast command selection. Fennedy et al.[6] also explored supporting soft keyboard hotkey with shortcut-akin mechanism for command execution.

3 System Design

Previous research reveals that sound propagated through the device would produces a stronger response than through the air[5]. We implemented a system to process real-time acoustic data captured by the device's built-in microphone to enable key-press finger contact types detection. The system pipeline is illustrated in Figure. 2. Please see the Appendix A for the implementation details.

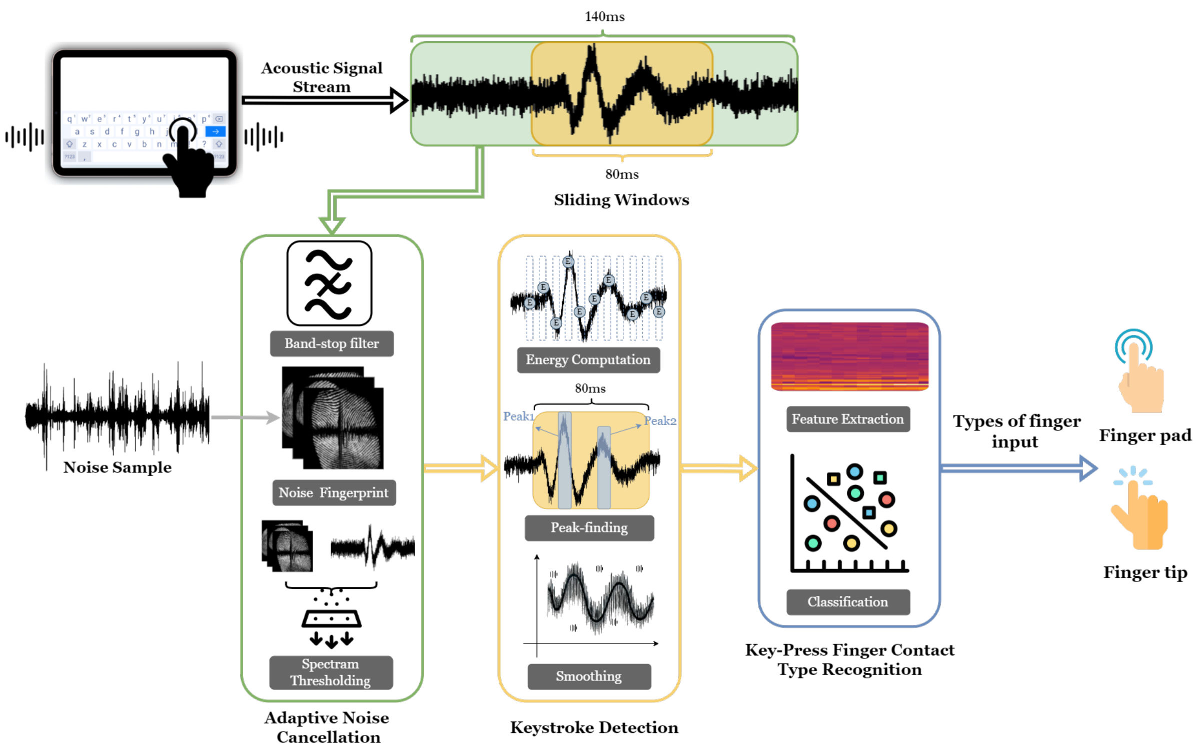

Figure 2: Illustration of the system pipeline. The pipeline mainly consists of three key component. The *Adaptive Noise Cancellation* component adaptively suppress background noise from the acoustic signal stream. The *Keystroke Detection* component separate valid keystroke segment from the stream. The *Key-Press Finger Contact Types Recognition* process the keystroke segment and predict the finger contact types.

3.1 Adaptive Noise Cancellation

During practice, we observed that background noise significantly influences system performance. To mitigate the background noise effect, we implemented a series of signal processing techniques on the captured keystroke acoustic signal. Previous research[27] has identified two main types of background noise in mobile keyboard use cases, namely white noise and ambient noise. To address this, we first applied a band-stop filter with a frequency stop range between 300Hz - 2kHz to remove white noise of each keystroke signal. We consider that given the mobility of tablet devices, the ambient noise would vary depending on the user's environment. To this end, we adopted an adaptive noise canceling algorithm to alleviate ambient noise based on spectrum thresholding[33]. Specifically, our system seamlessly captured a 1-second audio clip as the noise fingerprint once the keyboard was activated. We then performed a Short-time Fourier transform(STFT) on the noise fingerprint to compute the noise mean and standard deviation for each frequency component over time. The aforementioned statistical parameters are then used to establish a noise threshold for each frequency component THR_f in the keystroke signal:

$$THR_f = \mu_f + n\sigma_f \tag{1}$$

where μ_f and σ_f are the mean and standard deviation of the noise decibel scale, n denote number of standard deviations above mean. In practice, we set n to 1.5. To preserve more keystroke information, the mask is smoothed over time and frequency domains to avoid sudden changes before being applied to the input signal. The noise cancellation process was performed in the frequency domain. After that, we transform the filtered signal back to the time domain through inverse Fourier transform.

3.2 Keystroke Detection

Our system leverages the buffering technique to detect real-time keystroke acoustic signals. For every keystroke event detected from the touchscreen, we record both past and future frames from an audio buffer with size 6720 frames(140ms) under the sample rate of 48kHz in a sliding-window manner. Notably, keystroke events typically occur in rapid succession during typing. Previous research[28] has identified that the average keystroke interval during typing is around 100ms. Therefore, we set the window size to be less than 100ms to mitigate the window overlapping problem. We further conducted an experiment(Section. 4.3) to investigate the impact of window size on system performance. For every captured buffering window, we implement an adaptive peak-finding algorithm to further locate the keystroke hit peak. Specifically, we define a peak threshold by calculating the averaged sum of the mean and standard deviation in every window. We consider a typical keystroke should meet the following conditions: (i) The value of the keystroke peak

Table 1: Classification results of key-dependent model and key-independent model

	Key-Independent			Key-Dependent			Latency(ms)
	Accuracy	Recall	F1	Accuracy	Recall	F1	
KNN	0.927	0.938	0.929	0.935	0.941	0.935	90
SVM	0.922	0.941	0.924	0.924	0.957	0.929	150
RF	0.871	0.878	0.873	0.900	0.939	0.907	60
MLP	0.929	0.965	0.933	0.929	0.971	0.929	140
CNN	0.929	0.949	0.933	0.933	0.957	0.937	250

frame is the maximum within its buffer. (ii) The value of the keystroke peak frame must exceed the root mean square (RMS) value in the window by the threshold. (iii) The duration of the keystroke should exceed 60ms. (iv) The time intervals between the adjacent keystrokes should be at least 70ms. The keystroke segment was further smoothed by a Savitzky-Golay filter with the window size of 5.

3.3 Key-Press Finger Contact Types Recognition

As FT and FP impact produce different acoustic characteristics, we extract the acoustic feature in frequency domain for the recognition as previous researches on acoustic sensing suggested[5, 27]. To this end, we evaluated various acoustic features extraction techniques across multiple classifiers, including Mel-Frequency Cepstral Coefficients (MFCC), Fast Fourier Transform (FFT), and Short-Time Fourier Transform (STFT). We empirically chose MFCC which shows superior performance in terms of both latency and recognition accuracy for our task.

4 Evaluation

4.1 Data Collection

Through the word of mouth, we recruited 12 participants aged from 19 to 25(Mean=22.1, SD=2.08) from a local university for data collection. Six of them identified themselves as male and six as female. All of them were experienced tablet users. 11 of them were right-handed while 1 of them was left-handed. The study lasted 2.5 hours and participants received 50 RMB as compensation for their time. The study was conducted in an office where the average noise level is 42dB. Participants were provided a Samsung Tab S9 FE tablet with a custom data collection application running on it. The data collection application captures acoustic data in a sample rate of 48kHz. All participants were required to transcribe 100 phrases in FT and FP manner respectively. Half of those phrases were randomly chosen from a popular phrase set for text entry task[25], while we choose the other half of phrase from a holoalphabetic sentence set[1] to mitigate the frequency imbalance across different alphabet. During the study, we asked the participants to perform the typing as naturally as usual to avoid data bias. As a proof-of-concept, we only consider 36 character keys(i.e. letters and digits) and two most commonly used punctuations(i.e. periods and commas) in our research. As a result, we collected approximately 102,000 valid keystroke signal samples across 38 keys.

4.2 Offline Classification Performance

4.2.1 Key-Dependent Model and Key-Independent Model. In this experiment, we compared the performance of a set of machine learning classifiers on classifying key-press finger contact types on the collected dataset. We consider that the impact of the finger on different positions on the touchscreen produce distinct acoustic characterizes due to the sound propagation path differences. Therefore, we train and evaluate our method in both and key-independent scheme and key-dependent scheme on model performance. Specifically, for the key-independent scheme, we train one model to predict data from all keys. We form a testing set that averagely consists of 4500 samples for each key, while we mix the rest of data before splitting them into 8:2 ratio as a training set and validation set, respectively. For the key-dependent scheme, we train and test 38 key-dependent models for each key respectively by splitting data of every individual key into training, validation, and testing sets in an 8:1:1 ratio. We also experimented with the prediction latency of each candidate model. Table 1 summarizes the test accuracy, recall, and F1-score for each classifier using MFCC features. The result shows that the overall performance of the key-independent scheme was lower than that of the key-dependent scheme. Although Random Forest (RF) reaches the lowest latency during our offline experiments, it is suffering from the low accuracy problem, while K-Nearest Neighbors (KNN) achieves the best classification accuracy with a latency only 30ms greater than RF.

4.2.2 User profiling. We further conducted experiment on user profiling, where we train user-dependent models in key-dependent scheme. Specifically, We adopt a 4-fold data splitting scheme, where we randomly select 4 user data for testing, while we used the data from the rest of the users for training and validating. We repeated this process until all users were tested and averaged the performance for each classifier. The result was showed in Table. 2, where KNN outperforms the other method in terms of classification accuracy.

Taking consideration of both accuracy and latency, we chose the KNN model under the key-dependent training scheme for further analysis and experiments.

4.3 Effect of Window Size for Keystroke Detection

To investigate the optimal setting for real-time implementation, we conducted an experiment to evaluate the impact of window

[1]https://mseffie.com/assignments/calligraphy/Plethora%20of%20Pangrams.pdf

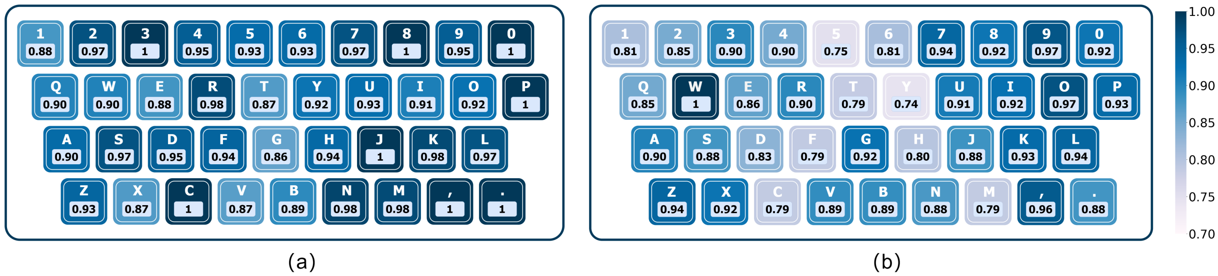

Figure 3: Online system performance. (a) Key-wise real-time classification accuracies in office area. (b) Key-wise real-time classification accuracies in restaurant area.

Table 2: Classification results of user-dependent models

	Accuracy	Recall	F1
KNN	0.963	0.970	0.963
SVM	0.948	0.966	0.950
RF	0.916	0.931	0.918
MLP	0.925	0.965	0.930
CNN	0.941	0.967	0.943

Table 3: Classification accuracy of different window size

rear / front	20ms	30ms	40ms	50ms
20ms	0.896	0.909	0.927	0.924
30ms	0.905	0.925	0.935	0.933
40ms	0.925	0.924	**0.938**	0.933
50ms	0.917	0.922	0.936	0.934

size for keystroke detection on system performance. To this end, we adopted a grid-search strategy by iteratively test through all combinations of the front frame and rear frame of the window. Consider the average keystroke interval during typing is averagely 100ms[28], we test both front frame and rear frame in 20ms, 30ms, 40ms, and 50ms. We run the experiment on our dataset using KNN model, and the classification accuracy is showed in Table.3. The result shows that the combination of 40ms front frame and 40ms rear frame achieved the best performance.

4.4 Online Performance

We further conducted a between-subject user study to evaluate the real-time performance of our system, particularly to experiment with the real-time detection accuracy in both quiet and noisy environment. To this end, we invited 6 participants aged from 23 to 25 (Mean=24.2, SD=0.68) from a local university and randomly divided them into two groups for two environments: 1) Office area (noise level = 44dB), 2) Restaurant area (noise level = 66dB). Each participant in both group was required to transcribe 14 random holoalphabetic sentences and 6 random number sequences in both FT and FP mode, respectively. Our system predicts the contact types in real-time and stored the results locally. We averaged out the accuracy in both FT and FP mode for every key under each environment, and showed the results in Fig.3.

The result shows that the average real-time key-wise recognition accuracy in quiet environment is 94%, while the accuracy drops to 88% when the environment goes noisy. Accuracy variances are also observed across keys. In particular, the accuracy for those keys that are close to both the left and right side are relatively higher than those that are in the middle. This could be caused by the affect of microphone location, where the two built-in microphones on the device are located on both the left and right side(landscape mode). Therefore, for keystroke event occurring close to either side would produce a larger signal to the corresponding side microphone, which would increase the recognition accuracy.

5 Usability Study on Text Formatting

To evaluate our system in a real-world usage scenario, we conducted a usability study focused on the text formatting task. We adopt our method to facilitate basic text formatting tasks (i.e. bold, italic, underline), where we allow users to switch between normal character input and formatted character input by changing their key-press finger contact types. A within-subject user study was conducted to compare the basic text formatting operation with the default system keyboard as a baseline condition and our method. Taking the bold task as an example, with the default system keyboard, users are required to manually select the target phrase and then choose the *bold* command from a pop-up menu. In comparison, with our method, the same task can be simplified by switching the finger contact types from FP to FT to apply the bold command during the text entry stage. We developed a demo application on the same devices we used for data collection, where we allowed users to switch between different text formatting modes through a floating menu.

5.1 Tasks and Procedure

We invited six participants to the study. Their age are between 24 and 27 (Mean=24.7, SD=1.24). Five participants identified them as male, and one identified as female. They reported their experience with text editing on tablet with touchscreens (average score=2.7/5, SD=0.75), smartphone touchscreen text editing experience (average score=3.2/5, SD=1.34), and physical keyboard text editing experience (average score=3.5/5, SD=1.12). The study was conducted in an office area. During the study, each participant is required to transcribe five mix-formatting sentences under two conditions: the baseline condition and ours, presented in a Latin-square counterbalanced order. We measured the total time spent on completing the task, and all participants were asked to fill out a System Usability Scale (SUS) survey after each condition. The detailed tasks and questionnaires are shown in the Appendix B.

5.2 Results

The average task completion time of our method is 283.58 seconds (SD=37.81), which is lower than the average task completion time of the baseline condition 328.46 seconds (SD=56.87). A Wilcoxon Signed Ranks Test shows that the task completion time of our method is significantly lower than the baseline condition (Z=2.024, $p<0.05$). We show the SUS results in Fig.4. The average total SUS score of our method is 35/45 (SD=6.08), and the average total SUS score of the baseline condition is 26.8/45 (SD=3.77). For the individual items of SUS questionnaire, Wilcoxon Signed Ranks Test shows that participants rate significantly high input speed (Z=2.041, $p<0.05$) and high functionality (Z=2.060, $p<0.05$) with our method then the baseline condition. One participant (male, 25 years old) explicitly mentioned that "*Your system is very interesting. Changing finger contact types is very easy for me, and I quickly learned to use this method.*" Another participant (female, 24 years old), while switching from our method to the baseline condition, commented that, "*I've already started to miss your system. Compared to this, changing finger contact types is more convenient and efficient.*"

6 Limitation and Future Works

Although our proposed system demonstrates promising results in detecting key-press finger contact types and shows potential applications in the text formatting task, there are several limitations that merit further investigation and improvement.

One of the limitation of our method is its relatively low recognition accuracy in noisy environments, especially for real-time implementation. Future work will focus on enhancing the robustness of the recognition system by collecting more training data in various of environment conditions or exploring more advance signal processing techniques. We also consider leveraging multi-modal sensor fusion techniques, such as integrating the IMU sensor data as an input feature to facilitate the recognition task. Moreover, as a proof of concept, we only implement and test our system on one device. Since the hardware configurations(e.g. microphone quality and location)on different device are various, we plan to deploy and evaluate our system on more devices.

In addition, our usability study shows potential on adapting our method on text formatting task in a mode switching manner. However, this still required providing an external menu for mode switching, which may disrupt the flow of typing during more complex text editing task. In the future, we would investigate more effective and intuitive manner to support more complex and comprehensive task with our method.

7 Conclusion

In this work, we demonstrated an innovative approach to augment the tablet typing experience by integrating key-press finger contact types as input. Our system captures real-time keystroke acoustic signals to distinguish between two finger contact types, namely finger tip and finger pad, to extend the input space of a standard touchscreen keyboard. We perform offline evaluation and online evaluation on our system, where the offline experiment shows that our system achieve a key-wise recognition accuracy of up to 96.3% and the online evaluation achieve an average recognition accuracy of 94% and 88% in quiet and noisy environment respectively. We further conduct a usability study on text formatting task, where we show that our method significantly outperform the default baseline method in terms of input speed.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China, Young Scientists Fund (Project No. 62402301) and the STU Scientific Research Initiation Grant (SRIG, Project No. NTF23024).

References

[1] Karan Ahuja, Paul Streli, and Christian Holz. 2021. TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 997–1009. https://doi.org/10.1145/3472749.3474801

[2] Hrvoje Benko, T. Scott Saponas, Dan Morris, and Desney Tan. 2009. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (Banff, Alberta, Canada) *(ITS '09)*. Association for Computing Machinery, New York, NY, USA, 93–100. https://doi.org/10.1145/1731903.1731924

[3] Xiaojun Bi, Shiri Azenkot, Kurt Partridge, and Shumin Zhai. 2013. Octopus: evaluating touchscreen keyboard correction and recognition algorithms via. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 543–552. https://doi.org/10.1145/2470654.2470732

[4] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual gesture keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 137–146. https://doi.org/10.1145/2380116.2380136

[5] Taizhou Chen, Lantian Xu, Xianshan Xu, and Kening Zhu. 2021. GestOnHMD: Enabling Gesture-based Interaction on Low-cost VR Head-Mounted Display. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2597–2607. https://doi.org/10.1109/TVCG.2021.3067689

[6] Katherine Fennedy, Sylvain Malacria, Hyowon Lee, and Simon Tangi Perrault. 2020. Investigating Performance and Usage of Input Methods for Soft Keyboard Hotkeys. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services* (Oldenburg, Germany) *(MobileHCI '20)*. Association for Computing Machinery, New York, NY, USA, Article 29, 12 pages. https://doi.org/10.1145/3379503.3403552

[7] Leah Findlater, Ben Lee, and Jacob Wobbrock. 2012. Beyond QWERTY: augmenting touch screen keyboards with multi-touch gestures for non-alphanumeric input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2679–2682. https://doi.org/10.1145/2207676.2208660

[8] Vittorio Fuccella, Poika Isokoski, and Benoit Martin. 2013. Gestures and widgets: performance in text editing on multi-touch capable mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2785–2794. https://doi.org/10.1145/2470654.2481385

[9] Hyunjae Gil, DoYoung Lee, Seunggyu Im, and Ian Oakley. 2017. TriTap: Identifying Finger Touches on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3879–3890. https://doi.org/10.1145/3025453.3025561

[10] Aakar Gupta, Muhammed Anwar, and Ravin Balakrishnan. 2016. Porous Interfaces for Small Screen Multitasking using Finger Identification. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 145–156. https://doi.org/10.1145/2984511.2984557

[11] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association

for Computing Machinery, New York, NY, USA, 59–70. https://doi.org/10.1145/2858036.2858052
[12] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279
[13] Ke He, Yongjie Duan, Jianjiang Feng, and Jie Zhou. 2022. Estimating 3D Finger Angle via Fingerprint Image. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 14 (March 2022), 22 pages. https://doi.org/10.1145/3517243
[14] Ke He, Chentao Li, Yongjie Duan, Jianjiang Feng, and Jie Zhou. 2024. TrackPose: Towards Stable and User Adaptive Finger Pose Estimation on Capacitive Touchscreens. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 161 (Jan. 2024), 22 pages. https://doi.org/10.1145/3631459
[15] Ana Beatriz Herthel and Anand Subramanian. 2020. Optimizing single-finger keyboard layouts on smartphones. *Computers & Operations Research* 120 (2020), 104947. https://doi.org/10.1016/j.cor.2020.104947
[16] Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, Min-Lun Tsai, Yen-Ting Yeh, Liwei Chan, Yi-Ping Hung, and Mike Y. Chen. 2014. TouchSense: expanding touchscreen input vocabulary using different areas of users' finger pads. 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, 189–192. https://doi.org/10.1145/2556288.2557258
[17] Zeyuan Huang, Cangjun Gao, Haiyan Wang, Xiaoming Deng, Yu-Kun Lai, Cuixia Ma, Sheng-feng Qin, Yong-Jin Liu, and Hongan Wang. 2024. SpeciFingers: Finger Identification and Error Correction on Capacitive Touchscreens. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 1, Article 8 (March 2024), 28 pages. https://doi.org/10.1145/3643559
[18] Xinhui Jiang, Yang Li, Jussi P.P. Jokinen, Viet Ba Hirvola, Antti Oulasvirta, and Xiangshi Ren. 2020. How We Type: Eye and Finger Movement Strategies in Mobile Typing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376711
[19] Tero Jokela, Jarno Ojala, and Thomas Olsson. 2015. A Diary Study on Combining Multiple Information Devices in Everyday Activities and Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3903–3912. https://doi.org/10.1145/2702123.2702211
[20] Jiwan Kim and Ian Oakley. 2022. SonarID: Using Sonar to Identify Fingers on a Smartwatch. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 287, 10 pages. https://doi.org/10.1145/3491102.3501935
[21] Sunjun Kim and Geehyuk Lee. 2016. TapBoard 2: Simple and Effective Touchpad-like Interaction on a Multi-Touch Surface Keyboard. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5163–5168. https://doi.org/10.1145/2858036.2858452
[22] Per Ola Kristensson and Shumin Zhai. 2007. Command strokes with and without preview: using pen gestures on keyboard for command selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1137–1146. https://doi.org/10.1145/1240624.1240797
[23] Huy Viet Le, Sven Mayer, and Niels Henze. 2019. Investigating the feasibility of finger identification on capacitive touchscreens using deep learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 637–649. https://doi.org/10.1145/3301275.3302295
[24] Xinshuang Liu, Yizhong Zhang, and Xin Tong. 2024. Touchscreen-based Hand Tracking for Remote Whiteboard Interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 71, 14 pages. https://doi.org/10.1145/3654777.3676412
[25] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971
[26] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting Hotkey use through rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 573–582. https://doi.org/10.1145/2470654.2470735
[27] 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
[28] Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.
[29] Keunwoo Park, Daehwa Kim, Seongkook Heo, and Geehyuk Lee. 2020. MagTouch: Robust Finger Identification for a Smartwatch Using a Magnet Ring and a Built-in Magnetometer. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376234
[30] Gulnar Rakhmetulla, Yuan Ren, and Ahmed Sabbir Arif. [n. d.]. GeShort: One-Handed Mobile Text Editing and Formatting with Gestural Shortcuts and a Floating Clipboard. 7 ([n. d.]), 1–23. Issue MHCI. https://doi.org/10.1145/3604259
[31] Gulnar Rakhmetulla, Yuan Ren, and Ahmed Sabbir Arif. 2023. GeShort: One-Handed Mobile Text Editing and Formatting with Gestural Shortcuts and a Floating Clipboard. *Proc. ACM Hum.-Comput. Interact.* 7, MHCI, Article 212 (Sept. 2023), 23 pages. https://doi.org/10.1145/3604259
[32] Jochen Rick. 2010. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 77–86. https://doi.org/10.1145/1866029.1866043
[33] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. 2020. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology* 16, 10 (2020), e1008228.
[34] Brian A. Smith, Xiaojun Bi, and Shumin Zhai. 2015. Optimizing Touchscreen Keyboards for Gesture Typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3365–3374. https://doi.org/10.1145/2702123.2702357
[35] Larry Tesler. 2012. A personal history of modeless text editing and cut/copy-paste. *Interactions* 19, 4 (July 2012), 70–75. https://doi.org/10.1145/2212877.2212896
[36] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. Association for Computing Machinery, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737
[37] Ying Yin, Tom Yu Ouyang, Kurt Partridge, and Shumin Zhai. 2013. Making touchscreen keyboards adaptive to keys, hand postures, and individuals: a hierarchical spatial backoff model approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2775–2784. https://doi.org/10.1145/2470654.2481384

A APPENDIX A: IMPLEMENTATION

We built a customized application on Samsung Tab S9 FE tablet with stereo built-in microphones located on the left side and the right side(landscape orientation) of the device for audio data streaming. The application captures the real-time acoustic signal in a sample rate of 48kHz and transmit the signal through TCP/IP protocol to a server in 140ms chunks. We implement a python server on a laptop PC with one RTX4060 NVIDIA GPU, 8GB RAM, and one Intel i7-13650HX CPU.

B APPENDIX B: USABILITY STUDY

B.1 Questionnaire

Fig.4 shows the detailed questionnaire results of our usability study. Each question can be rated from 0 to 5, of which 0 is completely not agree and 5 is completely agree.

B.2 The Sentence Transcription Tasks

Five mix-formatting sentences that we used for the usability study are shown in Fig.5, including two holoalphabetic sentences and three sentences from existing online documents.

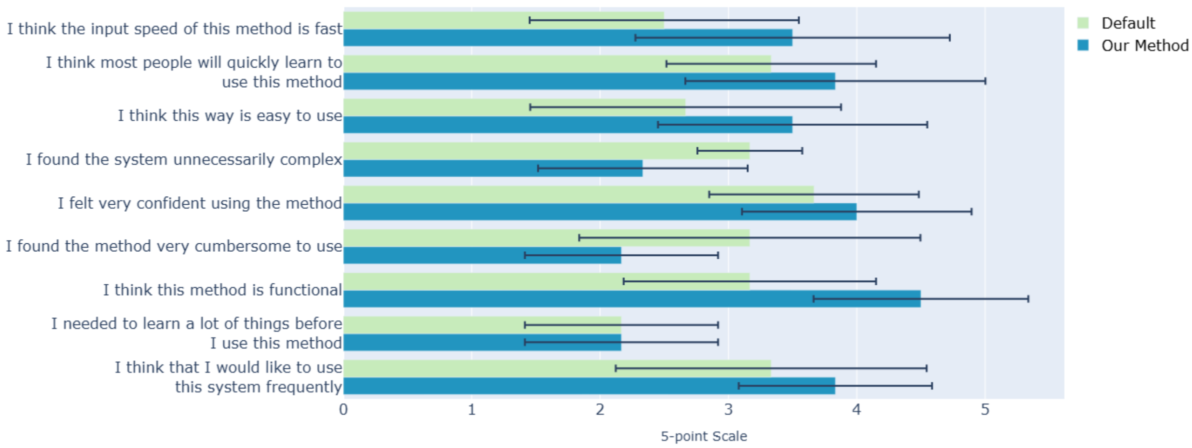

Figure 4: System Usability Scale results

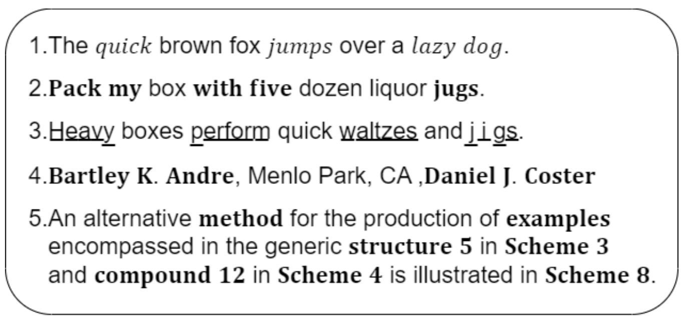

Figure 5: Sentence transcription task of the usability study