Yarnkey: A Thigh-worn Wearable Input Device for 2D Alphabetic Patterns using Conductive-threads-based Touch Sensing

KEN SHIBATA[1] KOYA NARUMI[2]

Abstract: In recent years, wearable text input devices using watches, glasses, and clothing have become popular. However, most of them have limitations related to posture and appearance when considering ubiquitous "anytime, anywhere" use. In this paper, we propose a wearable character input device, Yarnkey, which uses a conductive thread sewn into the thigh region of pants to recognize two-dimensional patterns corresponding to letters of the English alphabet, using six-point touch sensing. The advantages of Yarnkey include: it can be used with one hand, in a seated or standing position, is unobtrusive, small, and lightweight, and can be learned in a short period of time. This paper describes the implementation of Yarnkeyand details a user study conducted to validate the above advantages.

1. Introduction

With the spread of mobile devices such as smartphones, the number of computing devices around us has dramatically increased. At the same time, wearable input devices using objects that we can wear in our daily lives, such as watches, glasses, and clothes, are being used to realize ubiquitous computing, in which computers and interfaces are "woven" into everyday objects [1]. On the other hand, there are still some problems in existing research cases and products related to wearable input devices：

1. Use of both arms: Some arm-worn devices such as smartphones and the Levi's Trucker Jacket [2] require both arms to use and therefore is inefficient to use.
2. Limitations on posture: Arm-worn devices and gesture input devices such as the HoloLens [3] may not be optimally used in restricted environments such as crowded trains.
3. Visual invasion, weight, and volume: Head-mounted displays and similar devices are visually invasive. Also, devices with large keyboards, etc. are heavy and bulky and therefore are not optimal for being worn for a long amount of time.
4. Steep learning curve：Using non-traditional input methods (e.g. not a keyboard or not handwriting) results in a steeper learning curve.

Several methods have been proposed in the past to reduce these problems. For example, in TipText [4], a thin-film touch sensor is wrapped around the fingertip and input is made by tapping the fingers together. While this device is lightweight and easy to wear, it has the disadvantage of being visually invasive and blocking fingertips even when input is not required. In addition, Katayama et al. proposed a method of inputting data by attaching a keyboard divided into left and right sections to the thighs of both sides of the body [5]. However, even when this method is used, there is a large difference in appearance because the keyboard is attached to the clothing as it is. In addition, since the hard and heavy keyboard is attached to clothing, the user's experience of wearing the keyboard is considered to be very different from that of wearing a normal garment. In TelemetRing [6], a powerless coil is attached to each finger as a ring-type device, and changes in the magnetic field caused by the tapping motion are detected by the bracelet-type coil. With this method, the weight and size of the fingertip can be reduced because there is no need to attach a battery to the side of the ring, however the burden on the user remains because wearing a bracelet-type device with a certain weight and size is mandatory. Finally, Shinoda et al. have proposed a method for handwritten character input using a 1D capacitive touch sensor array fabricated by embroidery of conductive threads [7]. This method has the advantage of being lightweight and unobtrusive because the sensors can be directly embroidered and mounted on pants. However, due to the nature of detection by mapping to 1D input, this results in a steeper learning curve.

Therefore, this paper proposes Yarnkey, a wearable device that uses the thigh area of pants to input 2D patterns corresponding to letters of the alphabet. Yarnkeyuses 6-point resistive touch sensing with a conductive thread sewn into the pants to recognize 30 different hand-drawn patterns.

Yarnkeyhas the following advantages:

1. Only one finger is required to use the device.
2. Since putting hands on the thighs are possible in many

---

[1] William Lyon Mackenzie Collegiate Institute
[2] Graduate School of Engineering, The University of Tokyo

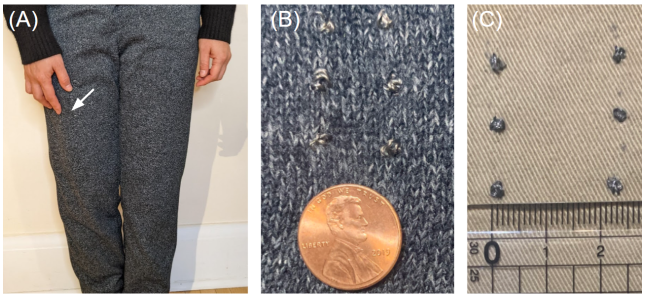


Fig. 1 (A) Overall view of Prototype 1 and (B) enlarged view of the sensor portion. (C) Enlarged view of Prototype 2 (fabric is different from Prototype 1).

circumstances, it can be used in extreme conditions. In a preliminary experiment, the author actually used Yarnkeyin a bus with a lot of vibration, and the input was almost the same as under normal conditions.

3. Since this uses resistor-based touch-sensing, the sensor shown on the surface is around $2mm^2$. It is therefore inconspicuous.
4. Since the patterns are similar to alphabets, it is easier to learn.

Subsequent chapters will detail the implementation of Yarnkey. We then report the results of quantitative and qualitative user experiments conducted to verify the effectiveness of our proposal. Finally, we conclude the paper with potential future improvements.

2. Design and Implementation

2.1 Design

As shown in Fig. 1, Yarnkeyrecognizes single-stroke letters similar to the alphabet by touch-sensing six points with conductive threads embroidered on the thighs of the garment. Let us consider a stroke that connects multiple touch sensor points to express the English alphabet, and consider the grid size required to input the character. For example, if a stroke connecting two to four points is detected on a 2×2 grid, ${}_4P_2 + {}_4P_3 + {}_4P_4 = 60$ patterns can be represented, enough to distinguish 26 letters in the English alphabet. However, with a 2×2 grid, the input strokes may have a pattern that is very different from the alphabet, and the learning cost regarding the correspondence between the strokes and the alphabet is expected to be high. Therefore, in this paper, we decided to use a touch sensor in the form of a 2×3 grid with two horizontal points and three vertical points. The grid corresponds to a 2D stroke pattern similar to the alphabet shown in Fig. 2.

2.2 Touch-Sensing Hardware

Fig. 3 shows the how the resistor-based touch sensing operates. First, when the finger is not in contact with point A, the voltage at point A is pulled up by a resistor to V_{DD}. When a finger touches point A, a contact resistor R_2 is inserted between AB, and the voltage at point A drops. By detecting this voltage drop with the AD converter built into the micro controller (Adafruit ItsyBitsy M4 Express), the micro controller could sense when a finger was touching the sensor.

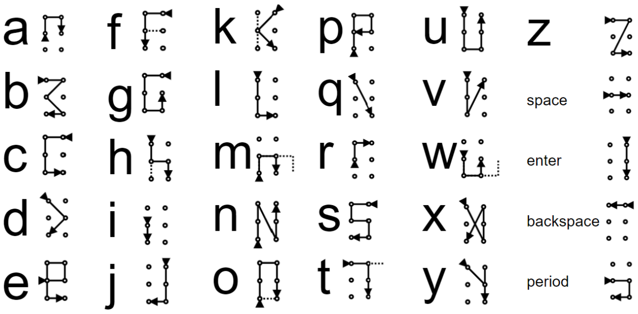


Fig. 2 2D recognition patterns based on the English alphabet.

In this paper, considering that the contact resistance between the human body and the sensor is at most several MΩ, the resistance R_1 is set to 20 MΩ. As for the capacitance C, a large value is desirable in order to reduce inductive noise of 60 Hz. However, if the capacitance is too large, the time constant of voltage change on contact becomes large, and strokes input at a speed above a certain level cannot be detected. Considering those two points, the capacitance C was set to be 1 nF.

Since direct contact between point A and the thighs can cause malfunctions, plastic tape is applied to the back of point A to reduce this in this paper.

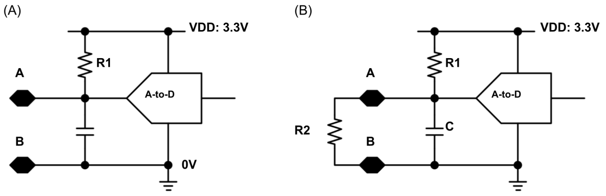


Fig. 3 rcuit configuration. Circuit model of a resistive touch sensor when (A) a finger is not touching and (B) it is touching.

2.3 Software Implementation

The micro controller is continuously measuring the output of 6 ADCs (analog digital converter). When the output of a point is over the threshold, it recognizes that the point was pressed and inserts it in a set of pressed points. When there is no interaction (no presses) for a certain time (”timeout”), the micro controller assumes that the input is finished and sends the corresponding key to the computer.

Commonly mistaken patterns are corrected in advance. The optimal timeout is different for each user. If it is too short, then the micro controller will stop the input before it was fully drawn. If it is too long, the device cannot be used as quickly. One improvement that can be made is to automatically calibrate the timeout.

In the current implementation, the timeout time is determined manually for each user. This is because the optimal timeout time varies depending on the user’s input speed. For example, if the timeout period is too short, the character input period ends in the middle of writing, and if it is too long, the user will not be able to type characters quickly.

Therefore, in the experiment described below, we adjusted the time between 0.5 s and 0.7 s according to the user.

2.4 Operation

Fig. 4 shows the input of the string “hello” from Yarnkeyto a smart phone. As can be seen, the input can be done with fine movements of the fingertip.

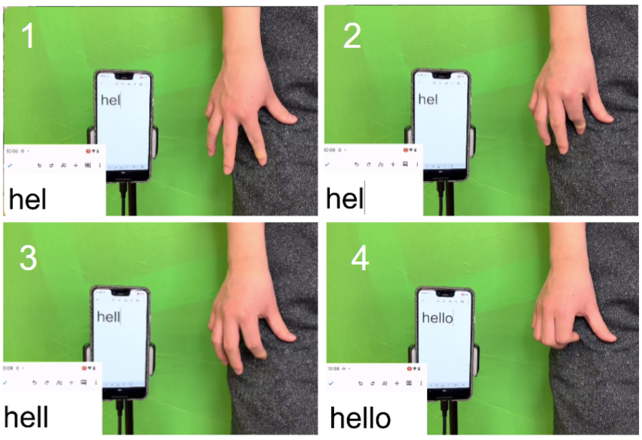


Fig. 4 The input of “hello” from Yarnkeyto a smartphone.

Fig. 5 also shows how text is input to the smartwatch when the Yarnkeyis worn on the arm rather than the thigh. Having an external input device for a small screen can be used to prevent occlusion of input and output.

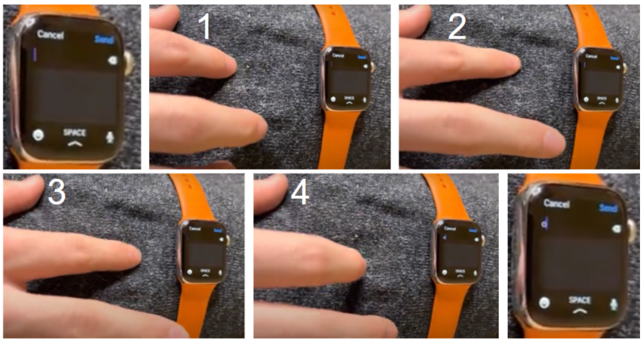


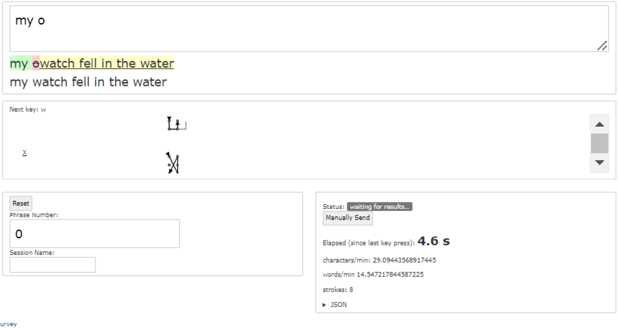
Fig. 5 The input of “ok” from Yarnkeyto the smartwatch.

3. Experiments and Evaluations

This chapter describes user tests conducted to verify Yarnkey’s input speed, error rate, learning speed, changes in input experience due to posture, differences in error rate due to input patterns, and ease of use.

3.1 Applications for User Testing

First, a web application shown in Fig. 6 was created using Svelte [8] for use in user testing. The application displays example sentences that the user should input and returns output that is color-coded according to the correctness or incorrectness of the user’s actual input. We used this application for user training and data acquisition. The example sentences displayed were randomly selected from the phrase set [9] developed by Mackenzie et al. for evaluating the performance of text input methods. The acquired data were sent to a server for storage and later analysis.


my o

my ~~o~~watch fell in the water

my watch fell in the water

Next key: w

x

Phrase Number: 0

Session Name:

Status: waiting for results...

Elapsed (since last key press): 4.6 s

characters/min: 29.09443568917445

words/min 14.547217844587225

strokes: 8

Fig. 6 Web application for user testing.

3.2 Procedures and Conditions

The experiment was conducted in Ontario, Canada, using the following procedure：

1. Explain Yarnkeyand how to use it in 5 minutes.
2. Ask participants to sit down.
3. Have participants learn how to use Yarnkey(15 minutes).
4. Measure the number of characters that can be entered and the error rate. (**Test #1**)
5. Have participants practice using Yarnkey(15 minutes).
6. Measure the number of characters that can be entered again and the error rate. (**Test #2**)
7. Perform the same measurements in a standing position. (**Test #3**)
8. Ask them to fill out a survey about the following items
   - System Usability Scale (SUS) [10](Ease-of-use rating, on a 5-point scale: 1-very much disagree, 3-don’t know, 5-very much agree)
   - Gender (male, female, other, don’t want to answer), age
   - Do you usually use a keyboard, and if so, what kind of device (computer or smartphone) and how often?

This procedure was performed on 5 participants (18 to 64 years old, average 47 years old, 2 males, 3 females, 0 others).

3.3 Evaluation Results

3.3.1 Input speed and error rate

Fig. 7 and Fig. 8 show Yarnkey’s input speed and error rate from user testing. First, as shown in the input speed results in Fig. 7 the average input speed was 0.37 char./sec and 0.46 char./sec for Test #1 and Test #2. According to [11], the average speed of handwritten input by a second grader is 0.40 char./sec. Compared to handwriting, the input speed of Yarnkeyis 93% of the average of second grade elementary school students in Test #1 and 115% in Test #2, which means that Yarnkeycan reach a level equivalent to the input speed of second grade handwriting in a total usage time of about 40 minutes.

In Test #2, in which about 15 minutes of practice was given after Test #1, the average input speed improved by

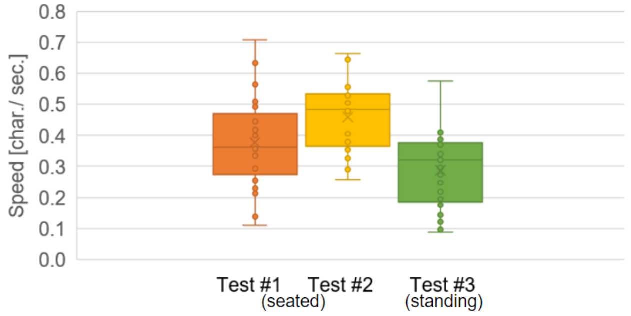


Fig. 7 Input speed ($n = 5$). Error bars are maximum and minimum values (excluding outliers). Boxes above and below are the 75% and 25% percentiles.

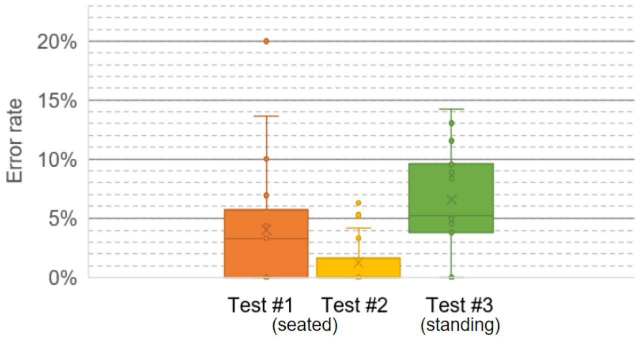


Fig. 8 Error rate ($n = 5$). Error bars are maximum and minimum values (excluding outliers). Boxes above and below are the 75% and 25% percentiles.

24% (= 0.46/0.37), indicating that the input speed improved even with a short practice time and that initial learning was somewhat easy. On the other hand, the input speed decreased to 0.29 char./sec in Test #3, where the input was performed in an upright state. This is thought to be due to the difficulty in visually recognizing the location of the sensing points and the unevenness of the fabric area where the sensor is located. Nevertheless, the results showed that it is possible to input at a speed somewhat close to that of a seated person even in a standing position.

In addition, the input error rate shown in Fig. 8 decreased from an average of 4% in Test #1 to an average of 1% in Test #2, confirming that learning is possible in a short time period even in this result. On the other hand, the average error rate in the standing state (7%) increased more than in the seated state. This is thought to be due to the difficulty of visual inspection and the unevenness of the cloth as well as the input speed discussion.

3.3.2 Input pattern

We then asked two of the five users to perform an additional experiment to determine the error rate for each letter. Table 1 shows a ranking of the error rates after asking users to enter a total of 51 patterns corresponding to all letters of the alphabet except SPACE. This shows, for example, that the error rates for R, Y, V, and S are relatively high. The reasons for the high error rates for these characters may be due to the following factors：

- The direction in which the pattern for letter R is written differs from the direction in which the letter R is hand written (when writing the letter R, the letter R

Table 1 A ranking of error rates for each letter of the alphabet, where Target indicates the number of times the letter was to be entered and Actual indicates the number of times it was entered.

Rank	Character	Error Rate	Target	Actual
0	a	0.0 %	51	51
1	c	0.0 %	51	51
2	d	0.0 %	51	51
3	e	0.0 %	51	51
4	f	0.0 %	51	51
5	i	0.0 %	51	51
6	j	0.0 %	51	51
7	k	0.0 %	51	51
8	l	0.0 %	51	51
9	m	0.0 %	51	51
10	n	0.0 %	51	51
11	o	0.0 %	51	51
12	q	0.0 %	51	51
13	t	0.0 %	51	51
14	u	0.0 %	51	51
15	w	0.0 %	51	51
16	x	0.0 %	51	51
17	b	2.0 %	51	50
18	g	2.0 %	51	50
19	h	2.0 %	51	50
20	p	2.0 %	51	50
21	z	2.0 %	51	50
22	s	3.9 %	51	49
23	v	3.9 %	51	49
24	y	3.9 %	51	49
25	r	7.8 %	51	47
26	*space*	NaN	0	0

is written from top to bottom and then to the upper right, while the R pattern is written from bottom to upper right).

- Y includes diagonals, and when writing it, surrounding dots may be touched and misidentified as T or other patterns.
- V requires writing the longest diagonal line, which may be more difficult.
- S has three fine bends, and when writing it, the surrounding dots may be touched and misidentified as different patterns.

Based on the above considerations, the following three points were considered for future improvement of input patterns: (1) make the pattern more intuitive to the user by matching the actual character and stroke direction, (2) minimize patterns that include diagonal lines and simplify non-diagonal lines, and (3) take into account variations in finger thickness when designing sensor size.

3.3.3 Usability

Table 2 summarizes the results of the post-experiment questionnaire. It shows that Yarnkeyreceived an average score of 4 or higher for all questions except Q1 (and less than 2 for questions where a lower score is desirable). In particular, Yarnkeyobtained good results for Q3 (ease of use), Q7 and Q10 (ease of learning), and Q8 (small size), indicating that Yarnkeymeets the target requirements. Note that P3 scored 1 for Q1 ("I think that I would like to use this keyboard frequently."). When P3 was interviewed about this point, it was found that this is because P3 does not use the computer at all when he is not at his desk and does not feel the need to use Yarnkey.

> I am almost never in a situation where I want to interact with my computer but am not at my desk. That is why I, personally, do not believe I would use it.

Table 2 Questionnaire questions and distribution of scores for each question $(n = 5)$。

		# of scores					Ave.
Question		**1**	**2**	**3**	**4**	**5**	**score**
Q1	I think that I would like to use this keyboard frequently.	1	0	1	2	1	3.4
Q2	I found the keyboard unnecessarily complex.	3	2	0	0	0	1.4
Q3	I thought the system was easy to use.	0	0	0	2	3	4.6
Q4	I think that I would need the support of a technical person to be able to use this keyboard.	4	0	1	0	0	1.4
Q5	I found the various functions in this system were well integrated.	0	1	0	2	2	4.0
Q6	I thought there was too much inconsistency in this system.	1	3	1	0	0	2.0
Q7	I would imagine that most people would learn to use this keyboard very quickly.	0	0	0	2	3	4.6
Q8	I found the keyboard very cumbersome to use.	2	2	1	0	0	1.8
Q9	I felt very confident using the keyboard.	0	1	0	2	2	4.0
Q10	I needed to learn a lot of things before I could get going with this system.	3	1	1	0	0	1.6

3.3.4 Other Considerations

Other issues identified in the user testing include the need to properly select the pull-up resistor R_1 of the touch sensor to suit the user. Since skin resistance varies greatly depending on the individual and the environment, it was necessary to adjust the pull-up resistor accordingly during the experiment. In the future, this could be improved by creating buffer circuits with individual components or by introducing calibration procedures on the software side.

Feedback from P3 also commented that Yarnkeymay be useful for field studies and game controllers used in public transportation, where natural behavior is required, due to its small external differences：

> One of the neat things about Yarnkeyis its stealth—you can imagine a ... field scientist finding that feature useful. ... I can also imagine it being very popular as a game controller, for use on a bus or train.

4. Future Works

4.1 Calibration function

As mentioned earlier, the timeout time and pull-up resistor R_1 used for recognition in this paper needed to be changed to suit the user. We are considering introducing a automatic calibration mechanism in the future.

4.2 Expansion of touch points

With the current implementation, the user must place his/her finger exactly on the sensor's point when inputting the data, and the alphabet pattern may deviate from the original shape due to the small number of touch points. Therefore, a possible future improvement would be to allow the user to start drawing from a larger area. In addition, this could increase the degree of freedom in the shape of the input pattern and allow for more natural input.

4.3 Improved usability and application to other uses

Other minor improvement ideas could include wireless communication between the micro controller and the main device (e.g. a mobile phone). In addition, there is a possibility that this can be used as a Braille input device by utilizing the common point pattern that a subset of Braille uses.

5. Conclusions

This paper proposes Yarnkey, a wearable device that uses a 6-point resistive touch sensor made of conductive thread to input 2D patterns resembling English alphabets. Yarnkeycan be used with a single finger, is discreet, small, and lightweight. User studies have shown that Yarnkeyis capable of inputting patterns with an average speed of more than 0.29 char./sec and an average error rate of less than 10 % even under seated and standing conditions. Furthermore, the input speed and error rate improved after a few minutes of training time.

Finally, demonstration videos, and an English version of this paper, etc. have been uploaded to the following URL [*1].

Acknowledgments This work was supported by JST Global Science Campus "Experts in Information Science (JST GSC EIS)" program hosted by the National Institute in Informatics, Information Processing Society of Japan, and

---

[*1] https://shibata.nyiyui.ca/yarnkey

the Japanese Committee for the International Olympiad in Informatics. I received much advice, inspiration, and support from the mentors and second-term students of the ”Experts in Information Science” program, as well as from Professor Kenichi Kawarabayashi and other masters of information science. Financial support was given by the Masason Foundation. We appreciate to Abby Kim and Kazuhiro Shinoda for their useful feedback.

References

[1] M. Weiser, “The computer for the 21st century,” *ACM SIGMOBILE mobile computing and communications review*, vol. 3, no. 3, pp. 3–11, 1999.
[2] “Jacquard by Google - Levi's®.” [Online]. Available: https://atap.google.com/intl/ja_jp/jacquard/products/levi-trucker/
[3] “Microsoft HoloLens — Mixed Reality Technology for Business.” [Online]. Available: https://www.microsoft.com/hololens/
[4] Z. Xu, P. C. Wong, J. Gong, T.-Y. Wu, A. S. Nittala, X. Bi, J. Steimle, H. Fu, K. Zhu, and X.-D. Yang, “TipText: Eyes-Free Text Entry on a Fingertip Keyboard,” in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 883–899. [Online]. Available: https://doi.org/10.1145/3332165.3347865
[5] T. Katayama, T. Terada, K. Murao, and M. Tsukamoto, “A Text Input Method for Half-Sized Keyboard Using Keying Interval,” in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2406367.2406375
[6] R. Takahashi, M. Fukumoto, C. Han, T. Sasatani, Y. Narusue, and Y. Kawahara, *TelemetRing: A Batteryless and Wireless Ring-Shaped Keyboard Using Passive Inductive Telemetry.* New York, NY, USA: Association for Computing Machinery, 2020, p. 1161–1168. [Online]. Available: https://doi.org/10.1145/3379337.3415873
[7] 篠田和宏, 原田珠華, 佐野由季, 安斉周, 矢谷浩司, “導電糸を用いたウェアラブルデバイス向け手書き文字入力デバイス,” in 電子情報通信学会総合大会, 2021.
[8] “Svelte • Cybernetically enhanced web apps.” [Online]. Available: https://svelte.dev/
[9] I. S. MacKenzie and R. W. Soukoreff, “Phrase Sets for Evaluating Text Entry Techniques,” in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 754–755. [Online]. Available: https://doi.org/10.1145/765891.765971
[10] “System Usability Scale.” [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
[11] L. Hamstra-Bletz and A. W. Blöte, “Development of handwriting in primary school: a longitudinal study,” *Perceptual and Motor Skills*, vol. 90, no. 3, Jan 1990.