Typing Braille Code in the Air with the Leap Motion Controller

Krzysztof Dobosz(✉) and Maciej Mazgaj

Institute of Informatics, Silesian University of Technology,
Akademicka 16, Gliwice, Poland
krzysztof.dobosz@polsl.pl, macimaz224@student.polsl.pl

Abstract. The paper presents the new concept of typing the Braille Code. Developed method use a Leap Motion controller to recognize fingers responsible for typing proper Braille dots in the air. The recognition of input characters bases on the analysis of coordinates of the hands, fingers, phalanges and hand joints in three-dimensional space, which is observed by the controller. The arrangement of the hands doesn't affect the accuracy of typed codes. Proposed method is implemented in the form of desktop application for personal computers. The evaluation shows the writing speed (7.4 WPM) and the MSD error rate (13.18%) comparable to Braille virtual keyboards using touchscreen.

Keywords: Braille code · Virtual keyboard · Text entry · Leap motion

1 Introduction

Reading and writing are fundamental human skills. Unfortunately, they may be very limited in the case of people with visual disability. Modern information technologies allow such people to overcome their own weaknesses. At the beginning, Braille monitors were used to read the text by touch the convex Braille code, and the Braille type-writers for writing. Development of technologies for mobile devices has opened new opportunities for assistance to the blind. Nowadays screen readers have integrated witch each smartphone, phablet and tablet. Hence, the reading of the textual content of presented information is not a problem. At present every mobile device has a touch screen that can be used as a virtual Braille keyboard. Multi-point touch sensors can recognize six finger typing Braille dots at the same time. The problem is to put both hands over the surface of small mobile device as a smartphone with diagonal up to 5 in. Although a lot of solutions to write Braille code on the small touch screen has already developed, most of them use fewer number of fingers, and the idea of *"one finger per one dot"* referring to the traditional Braille type-writers is problematic in the realization.

Therefore, new solutions to efficiently type in Braille are still awaited. Touch screens of mobile devices have several significant limitations: small size, low sensitivity in some conditions, recognized number of fingers in multi-touch mode, etc. To solve this problem the user can lift his fingers up. The aim of our studies were to develop a method for efficient typing Braille code in the air.

A. Gruca et al. (eds.), *Man-Machine Interactions 5*,
Advances in Intelligent Systems and Computing 659, DOI 10.1007/978-3-319-67792-7_5

2 Background

The classical Braille type-writer was the *Perkins Brailler*, which modern version was developed in 2008 [3]. This is mechanical device that consists of six keys corresponding to the six points of a Braille Code. Additionally the keyboard has also keys for space and backspace.

The idea of *"one finger per one dot"* was also adopted to the rapidly developed mobile technologies. That was easy for first portable phones with limited number of physical buttons. The *BrailleTap* [7] solution used a six buttons located in two columns: '2'-'5'-'8' and '3'-'6'-'9' to type proper dots of the Braille keyboard. In the age of touch screens, keys became virtual and occupied its subareas. In the *BrailleType* [13], the user marked required dots using only one finger by tapping proper buttons one by one. *LeBraille* [5] is the similar project improving the Braille typing by audio and vibration feedback. Two fingers were used in *TypeInBraille* [11], where two dots were type at the same time row-by-row. Another approach called *Perkinput* operated with three fingers, which are used at the same time [1]. First, the left column were entered, then the right one.

Apart from those solutions, also the idea of *"one finger per one dot"* is found for the technology of touch screens. One of interesting text-entry methods for visually impaired people was called *BrailleTouch* [14]. In that approach the smartphone has to be hold horizontally and with the touch screen facing away from the user. The application interface includes six virtual buttons corresponding to Braille dots. Each of the fingers used to type, corresponds to the same Braille dot as in the case of using a physical device.

The topic of the studies assumes the use of LeapMotion controller. The precision and reliability of the Leap Motion were analyzed using a high-precision optical tracking system [8] and research shown high potential of the device in human-computer interfaces. Research [19] has shown in realistic scenarios, that an accuracy is less than 0.5 mm for motions. While not as precise as more sophisticated optical motion capture systems, the Leap Motion controller is sufficiently reliable for the measurement of motor performance in pointing tasks that do not require high positional accuracy [18]. The accuracy of detection for static hands is below the human hand tremor [17].

The aim of another study [2] was to provide an approach that focuses on the horizontal axis of interaction only. Horizontal movements are commonly used independently of the interaction device (e.g., mouse, touch pad, touch screen). More degrees of freedom have to be controlled with the Leap Motion controller than with a mouse device, requiring advanced motor and coordination skills. This is reflected by the overall results of the error rate. It was three-times higher than the error rate achieved with a standard mouse device.

Researchers observed that people prefer related gestures for dichotomous tasks and more disagreement occurring for abstract tasks, such as "open browse" [20]. They defined several measures for Leap Motion gestures, such as gesture volume and finger-to-palm distance, which were used to evaluate gestures performed by research participants.

Due to its rather limited sensory space and inconsistent sampling frequency, it cannot currently be used as a professional tracking system [10]. However, it can be applied in many other useful touchless interfaces, i.e. for online virtual learning [9], or as a part of virtual reality technology to reproduce the ancient ruin [4]. However, the space monitored by Leap Motion controller, can accommodate two hands typing virtual keys. Some researchers have tried to design a universal interface for all to enter text using 10 fingers [12]. Unfortunately, that concept is still without the implementation and evaluation.

3 Method

3.1 Assumptions

The Leap Motion undoubtedly is a noticeable input device for gesture-based human-computer interaction. This controller is a small peripheral device. It uses two monochromatic IR cameras and three infrared LEDs, observes almost hemispherical area in a distance about one meter. The Leap Motion must be placed facing upward (Fig. 1).

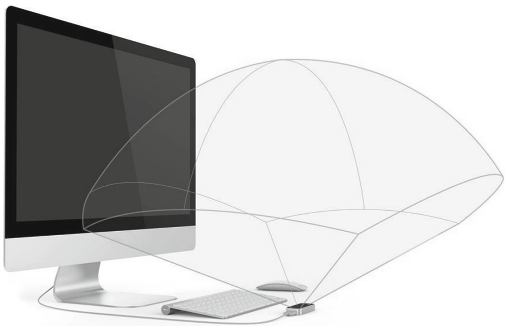
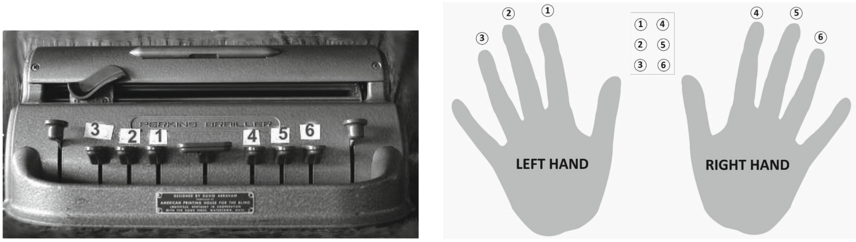


Fig. 1. Observed area of the Leap Motion controller [6]

For the development of the Braille code entry method, the following assumptions were considered:

- the interaction always use both human hands,
- hands need not be located in the same plane,
- the position of the hands in three-dimensional space should not affect the accuracy of typed codes,
- the recognition of input characters in Braille code should be based on the analysis of the coordinate position of the hands, fingers, phalanges and hand joints.



(a) typewriter (b) Hands

Fig. 2. The assignment of fingers to Braille dots

The typing of a single character in the Braille code is performed by the fingers of both hands in the same manner resembling fundamental gesture “key tap” as in the case of using a hardware type-writer (Fig. 2).

Next, following concepts are introduced:

– hand plane P_H,
– base plane P_B,
– finger collision with P_B.

3.2 Analysis

To determine the hand plane in three-dimensional space, it is enough to find three non-collinear points that belong to this hand. The Leap Motion controller provides us with information about these points. The first point H_1 is the center of the palm. The second point H_2 is the position of the joint connecting the metacarpal bone of the phalanx of the index finger. The third point H_3 is the position of the joint connecting the metacarpal bone of the proximal phalanx of the little finger (Fig. 3). For non-collinear points H_1, H_2, H_3 there is only one plane P_H fulfilling the determinant equation.

$$P_H : \begin{vmatrix} x & y & z & 1 \\ x_{H1} & y_{H1} & z_{H1} & 1 \\ x_{H2} & y_{H2} & z_{H2} & 1 \\ x_{H3} & y_{H3} & z_{H3} & 1 \end{vmatrix} = 0 \tag{1}$$

Now, for the hand plane defined by formula (1) must be defined the base plane P_B, which is parallel to the P_H, and away from it a given distance d. In order to determine P_B, first should be defined:

$$\overrightarrow{w_d} = \overrightarrow{a} \times \overrightarrow{b} = [y_a z_b - z_a y_b;\ z_a x_b - x_a z_b;\ x_a y_b - y_a x_b] \tag{2}$$

where $\overrightarrow{a}$ and $\overrightarrow{b}$ are vectors of anchorage point H_1:

$$\overrightarrow{a} = [x_{H2} - x_{H1};\ y_{H2} - y_{H1};\ z_{H2} - z_{H1}]$$
$$\overrightarrow{b} = [x_{H3} - x_{H1};\ y_{H3} - y_{H1};\ z_{H3} - z_{H1}]$$

To vector calculated from the formula (2) can be used as a normal vector after the normalization. Then, using given distance d between the planes, we can determined points belonging to P_B:

$$B_1 = H_1 + \frac{\overrightarrow{w_d}}{\|\overrightarrow{w_d}\|}d$$
$$B_2 = H_2 + \frac{\overrightarrow{w_d}}{\|\overrightarrow{w_d}\|}d \quad (3)$$
$$B_3 = H_3 + \frac{\overrightarrow{w_d}}{\|\overrightarrow{w_d}\|}d$$

The plane determined by the points B_1, B_2, B_3 (3) and the line passing through the points F_1 and F_2 (finger points) intersect in the point I, which can be calculated by solving the equations:

$$\begin{cases} x_I = x_{F1} + (x_{F2} - x_{F1})t \\ y_I = y_{F1} + (y_{F2} - y_{F1})t \\ z_I = z_{F1} + (z_{F2} - z_{F1})t \end{cases}$$

where

$$t = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_{B1} & x_{B2} & x_{B3} & x_{F1} \\ y_{B1} & y_{B2} & y_{B3} & y_{F1} \\ z_{B1} & z_{B2} & z_{B3} & z_{F1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 0 \\ x_{B1} & x_{B2} & x_{B3} & x_{F2} - x_{F1} \\ y_{B1} & y_{B2} & y_{B3} & y_{F2} - x_{F1} \\ z_{B1} & z_{B2} & z_{B3} & x_{F2} - z_{F1} \end{vmatrix}} \quad (4)$$

Next, the distance between points F_1, F_2 and I be calculated.

$$|F_1F_2| = \sqrt{(x_{F2} - x_{F1})^2 + (y_{F2} - y_{F1})^2 + (z_{F2} - z_{F1})^2}$$
$$|F_1I| = \sqrt{(x_I - x_{F1})^2 + (y_I - y_{F1})^2 + (z_I - z_{F1})^2} \quad (5)$$
$$|F_2I| = \sqrt{(x_I - x_{F2})^2 + (y_I - y_{F2})^2 + (z_I - z_{F2})^2}$$

The intersection point I belongs to the line segment $|F_1F_2|$ when

$$|F_1F_2| \geq max(|F_1I|, |F_2I|) \quad (6)$$

All points, planes and vectors mentioned in this section are illustrated in the Fig. 3.

Such calculation should be provided for each of six fingers corresponding to Braille dots. Let's remember that we have two independent hand planes. For each of them we have another base plane.

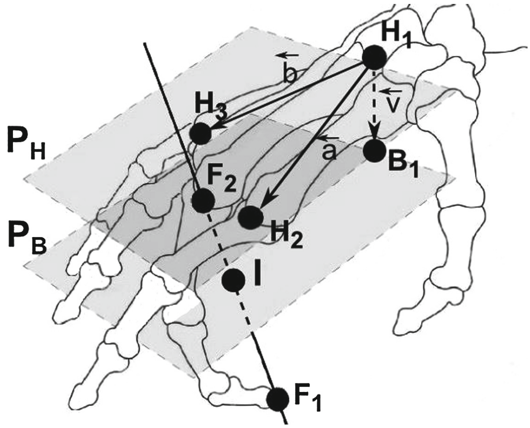


Fig. 3. Hand typing Braille code in the 3D space.

4 Evaluation

For needs of the evaluation phase, the proposed method was implemented in a form of desktop application for personal computer (Fig. 4). The graphical interface of the application allows the user to set the distance d between the base plane and the hand plane, given in millimeters. Second parameter is the number of skipped frames by the controller until all user's fingers take correct positions and hand is stabilized. This research software tool automatically measures the typing speed and its accuracy (number of errors). Every typed letter i read by voice in the application.

Six visually impaired participants (five male, one female) in the age range 22–47 were involved in the evaluation phase. None of them was completely blind. No one was also perfectly familiar with reading and writing in Braille, so the assistant sometimes supports them suggesting the Braille dots for every character. Each participant started with a 10-min preliminary training. Next, he was supposed to write the pangram: *"the quick brown fox jumps over the lazy dog"*. The sentence consists of 9 words built of 43 characters including 8 white spaces. The distance between hand planes and base planes was set to 40 mm. The number of omitted frames was set to 10.

Table 1 presents results of the experiment (*time* - in seconds, *err* - number of incorrectly selected keys), when the users typed the pangram. The best time of test was equal 51 s, the worst - 98 s. Initially, despite the preliminary trials, the typing in the air was not easy for users (maximal number of mistakes was 14), then they were more familiar typing faultlessly. Calculated average text entry factor was $WPM = 7.4$, MSD error rate [15] equals about 13.2%. Considering only the last trial, when participants were more experienced, the results were better ($WPM = 7.76$, $MSD = 7.36$).

So far the approaches of typing the Braille code in the air have not been presented. Therefore, described method can be compared to virtual Braille

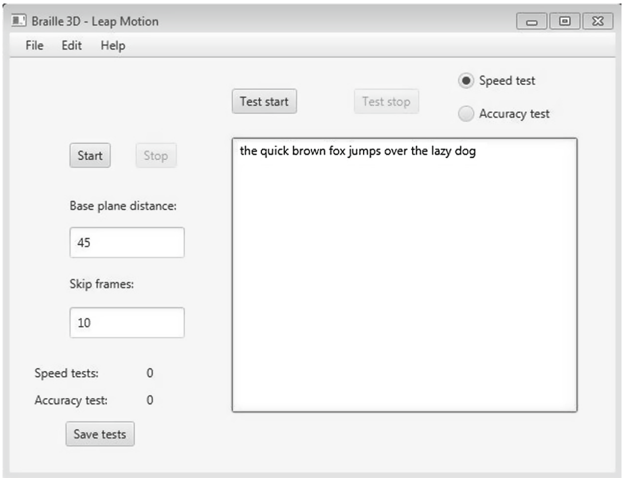


Fig. 4. Research tool

Table 1. Results of typing in the air.

Trial	User 1		User 2		User 3		User 4		User 5		User 6	
	time	*err*	*time*	err_k	*time*	*err*	*time*	*err*	*time*	*err*	*time*	*err*
1	96	14	73	8	66	8	78	12	69	5	81	7
2	98	10	76	4	62	5	72	10	61	1	69	8
3	77	11	72	5	61	4	71	6	54	0	74	3
4	74	10	67	2	54	4	66	9	51	2	72	3
5	81	6	63	0	62	1	67	7	55	0	71	5
avr	85.2	10.2	70.2	3.8	61	4.4	70.8	8.8	58	1.6	73.4	5.2

Table 2. Comparison to virtual Braille keyboards.

Name	No. fingers	WMP	MSD error rate
BrailleType	1	1.45	8.91%
Perkinput	3	6.05	3.52%
Braille in the air	6	7.4	13.2%
Braille in the air (best)	6	7.76	7.36%
BrailleTouch	6	23.2	14.5%

keyboards using the touch screen of mobile devices (Table 2). The obtained results locate the air approach in the middle of the list. The most similar solution is the *BrailleTouch* because it also represents the idea of *"one finger per one dot"*. It presents better typing efficiency [16] due to the special way of holding the phone, but its error rate is worse - especially in the case of last trial of typing in the air.

5 Conclusions

The Leap Motion controller has become very popular device lately because of its use for VR goggle needs. As we have shown it can be also used for blind people. Dots of the Braille code can be indicated by corresponding fingers.

The major advantage is the independence from the plane of typing. Both hands may be kept free in the air, although they must be in the space observed by the controller. This space is much wider than the size of mobile touch screens, so provides greater freedom of typing.

Obtained results are satisfactory. The typing speed is comparable to approaches using touch screens of mobile devices. Moreover, it can be improved by personalization that takes into account the user's fingers length (the distance between the palm and the base plane) and the speed at which relevant fingers taps dots in the air (proper number of skipped frames recorded by the controller).

The disadvantage of this solution is possible fast tiredness when typing long text. It would be a good idea to use hand support to palms in a proper distance above the controller (about 20–30 cm).

Acknowledgements. Publication financed by the Institute of Informatics at Silesian University of Technology, statutory research no. 02/020/BK-16/0077.

References

1. Azenkot, S., Wobbrock, J.O., Prasain, S., Ladner, R.E.: Input finger detection for nonvisual touch screen text entry in perkinput. In: ICGI 2012, Toronto, Ontario, Canada, pp. 121–129. Canadian Information Processing Society (2012)
2. Bachmann, D., Weichert, F., Rinkenauer, G.: Evaluation of the leap motion controller as a new contact-free pointing device. Sensors **15**(1), 214–233 (2014)
3. Copeland, D.: A lighter touch: Perkins school hopes compact Braille machine brings renewed interest. The Boston Globe (2008)
4. Dianxing, L.: Research on digital exhibition old summer palace somatosensory interaction based on leap motion. In: ICECEE 2015, Jinan, China. Advances in Computer Science Research, pp. 1079–1083. Atlantis Press (2015)
5. Façanha, A.R., Viana, W., Pequeno, M.C., de Borba Campos, M., Sánchez, J.: Touchscreen mobile phones virtual keyboarding for people with visual disabilities. In: HCI 2014, Heraklion, Crete, Greece, pp. 134–145. Springer (2014)
6. Fossbytes: Now control your PC with futuristic leap motion. https://fossbytes.com/now-control-pc-futuristic-leap-motion. Accessed Sept 2016
7. Guerreiro, T., Lagoá, P., Santana, P., Gonçalves, D., Jorge, J.: NavTap and BrailleTap: Non-visual texting interfaces. In: RESNA 2008, Washington, DC, USA. ACM (2008)
8. Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., Sodnik, J.: An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors **14**(2), 3702–3720 (2014)

9. He, L., Hu, X.Q., Wei, D.: The study of interaction about the online virtual learning community based on leap motion. In: ICCIS 2014, NanChong, China, pp. 1225–1229. IEEE (2014)
10. Jakus, G., Guna, J., Tomažič, S., Sodnik, J.: Evaluation of leap motion controller with a high precision optical tracking system. In: ICHCI 2014, Heraklion, Crete, Greece, pp. 254–263. Springer (2014)
11. Mascetti, S., Bernareggi, C., Belotti, M.: TypeInBraille: a Braille-based typing application for touchscreen devices. In: SIGACCESS 2011, Dundee, Scotland, UK, pp. 295–296. ACM (2011)
12. Mourouzis, A., Arfaras, G., Kilintzis, V., Chouvarda, I., Maglaveras, N.: Touchless text entry for all: initial design considerations and prototypes. In: UAHCI 2015, Los Angeles, CA, USA, pp. 37–49. Springer (2015)
13. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: Brailletype: unleashing Braille over touch screen mobile phones. In: Human-Computer Interaction-INTERACT 2011, pp. 100–107 (2011)
14. Romero, M., Frey, B., Southern, C., Abowd, G.D.: Brailletouch: designing a mobile eyes-free soft keyboard. In: HCIMDS 2011, Stockholm, Sweden, pp. 707–709. ACM (2011)
15. Soukoreff, R.W., MacKenzie, I.S.: Measuring errors in text entry tasks: an application of the levenshtein string distance statistic. In: CHI 2001, Seattle, WA, USA, pp. 319–320. ACM (2001)
16. Southern, C., Clawson, J., Frey, B., Abowd, G., Romero, M.: An evaluation of brailletouch: mobile touchscreen text entry for the visually impaired. In: HCIMDS 2012, San Francisco, CA, USA, pp. 317–326. ACM (2012)
17. Sturman, M.M., Vaillancourt, D.E., Corcos, D.M.: Effects of aging on the regularity of physiological tremor. J. Neurophysiol. **93**(6), 3064–3074 (2005)
18. Tung, J.Y., Lulic, T., Gonzalez, D.A., Tran, J., Dickerson, C.R., Roy, E.A.: Evaluation of a portable markerless finger position capture device: accuracy of the leap motion controller in healthy adults. Physiol. Meas. **36**(5), 1025 (2015)
19. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. Sensors **13**(5), 6380–6393 (2013)
20. Zaii, I.A., Pentiuc, S.G., Vatavu, R.D.: On free-hand TV control: experimental results on user-elicited gestures with leap motion. Pers. Ubiquit. Comput. **19**(5–6), 821 (2015)