DasherVR: Evaluating a Predictive Text Entry System in Immersive Virtual Reality

Sebastian Pape
pape@vr.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Jan Jakub Ackermann
jan.jakub.ackermann@rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Tim Weissker
weissker@vr.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Torsten W. Kuhlen
kuhlen@vr.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

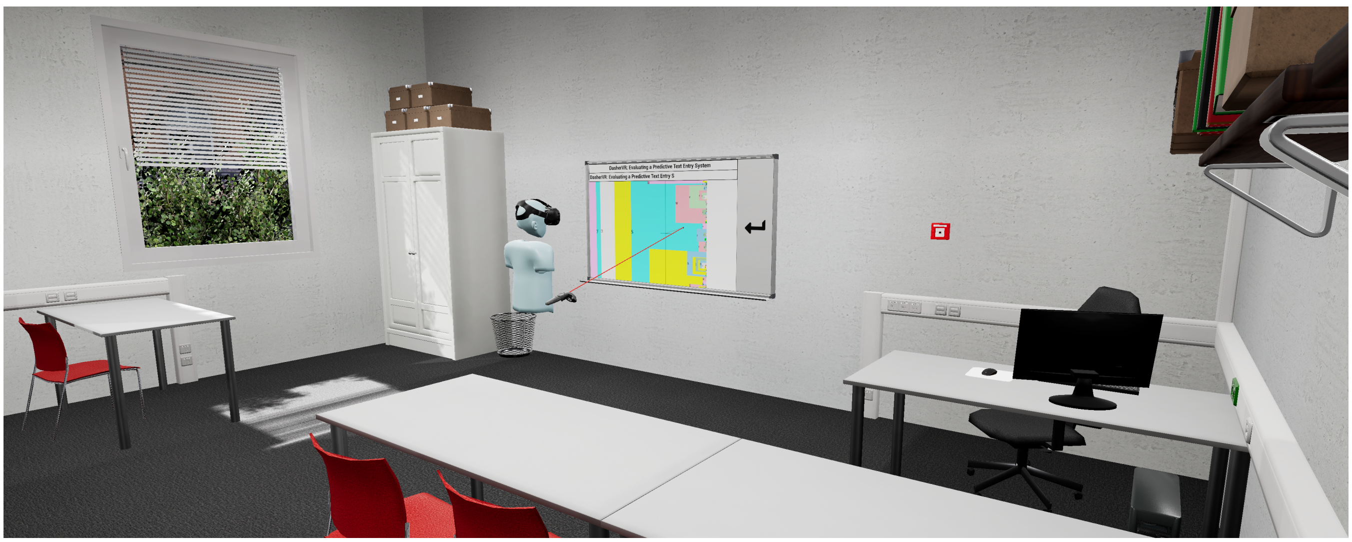

Figure 1: DasherVR operated by a user in the virtual environment that was used in the evaluation.

ABSTRACT

Inputting text fluently in virtual reality is a topic still under active research, since many previously presented solutions have drawbacks in either speed, error rate, privacy or accessibility. To address these drawbacks, in this paper we adapted the predictive text entry system “Dasher” into an immersive virtual environment. Our evaluation with 20 participants shows that Dasher offers a good user experience with input speeds similar to other virtual text input techniques in the literature while maintaining low error rates. In combination with positive user feedback, we therefore believe that DasherVR is a promising basis for further research on accessible text input in immersive virtual reality.

TIAM23, April 23, 2023, Hamburg, Germany

CCS CONCEPTS

• **Human-centered computing** → *Accessibility systems and tools*; **Virtual reality**; **Text input**; **Gestural input**.

KEYWORDS

Virtual Reality, 3D User Interfaces, Text Input, Dasher, Accessibility

1 INTRODUCTION

Text input is one of the major challenges in VR applications that is still not solved to a satisfactory level. While the most commonly used input metaphor remains a virtual copy of a physical keyboard in different shapes (e.g. [1, 2, 7, 22]), the transfer of real-world devices into a virtual environment with limited haptic feedback rarely works well. From an accessibility point of view, any form of text input in virtual reality should support the user in writing words and sentences both accurately and efficiently. Conventional keyboard designs, however, are not optimal in this regard as they require the user to precisely hit small keys as quickly as possible. In addition, conventional keyboard layouts also come with additional overheads due to the high redundancies of natural text.

One project that tackled the problem of text entry in desktop computing from a standpoint of information theory was the "Dasher" project [31]. The Dasher system allows users to input text in arbitrary languages and with arbitrary alphabets using continuous gestures rather than distinct keystrokes. Moreover, the system is based on text prediction that makes likely letters easier to select.

The original system turned out to be reliable and became well-known for motion-impaired users in desktop computing since its release. As the Dasher input system provides many input metaphors and rendering modes in its desktop user interface, we were confident that the system could be a helpful addition in virtual reality as well.
In this paper, we therefore report on our efforts of porting the Dasher system into a virtual environment in a straightforward manner. To get an impression of the advantages of our system, DasherVR, we performed a study to evaluate the user experience, as well as input speed and error rate. In summary, our work led to the following contributions:

- A license-change of the original Dasher library in cooperation with the authors as a prerequisite for its integration into modern game engines[1]
- Several modernizations and refactorings of the original Dasher library as well as the addition of a new build system to facilitate future developments
- The integration of the refactored library into a plugin for Unreal Engine 4, which is openly available to the public[2]
- The results of a user study with 20 participants, showing the applicability of the Dasher System for text input in VR

Our results encourage the usage of the Dasher system in VR, while revealing points for future work on improving and optimizing the system for general XR usage and improved user experience. While we chose to use a simple pointing and clicking metaphor in this work, alternative ways of interacting with the system in the future should be evaluated. These could include the usage of the analog-stick or touchpad of the controller, the eye tracking of modern head-mounted displays (HMDs) as well as finger tracking for controller-less usage to further increase accessibility.

2 RELATED WORK

There are many techniques that allow inputting text in virtual and augmented reality with different modalities, but often pose certain requirements on the users, which sometimes can't be fulfilled by motion-impaired users. Many of the techniques use both hands [11, 35, 36] and/or require high accuracy or finger dexterity for inputting text on virtual keyboards [1, 2, 22, 27], the user's hand [8, 33] or touchpads [11, 14, 18, 37]. Others use rather big controller movements like [3, 16] that can be fatiguing and require a big range of physical motion.
There exist techniques that involve gaze or eye tracking like [17, 20] or speech input like [4, 19] which in principle offer a high accessibility. While speech input can't be used for sensitive information and often doesn't work well due to a noisy environment, the usage of virtual keyboards in combination with eye or gaze tracking are often fatiguing [6].
The Dasher input metaphor was studied before in many variants and adaptations in desktop and mobile computing. These implementations involved either mouse, touchscreen or joystick input [31], eye/gaze tracking [29, 32] and even brain-computer interfaces [34]. An adaptation of Dasher which incorporates speech input was done by [30] and further adapted by [12, 13] to include motion tracking elements, eventually moving away from the original Dasher interface. Our work in this paper extends this body of related work by exploring the use of Dasher in immersive virtual reality with a 3D-tracked controller as the main input device.

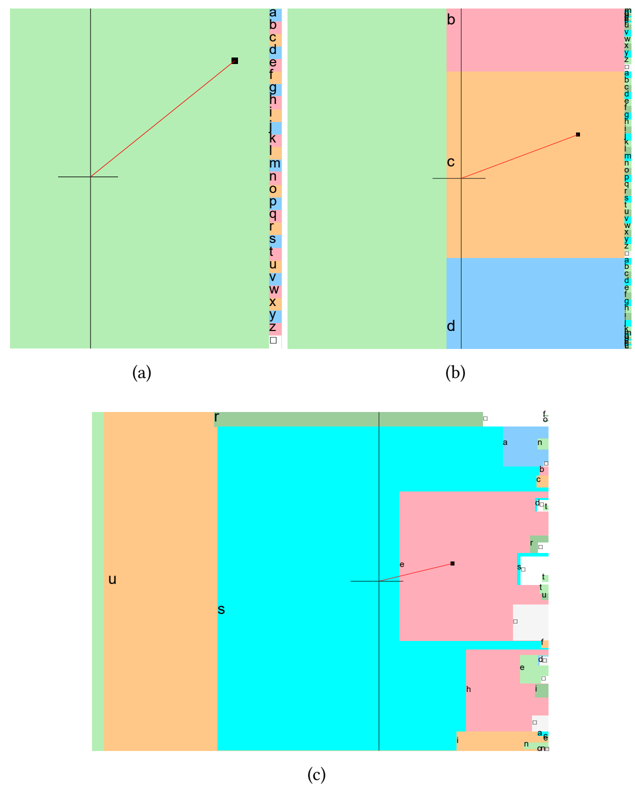

Figure 2: The Dasher interface, where (a) and (b) show an example with equal probabilities for each character. (a) shows the initial state, before choosing the first letter among all letters of the alphabet. (b) shows choosing the second letter after inputting "c" already. (c) shows an example with a trained language model, where probabilities are mapped to the height of each box. The state is shown after inputting "us" and getting the highest prediction for "use". Deeper predictions show "used", "user" and "uses". The white box □ denotes the space character.

[1]https://github.com/dasher-project/dasher-MIT
[2]https://git-ce.rwth-aachen.de/vr-vis/VR-Group/unreal-development/plugins/dashervr

3 DASHERVR

In this section we will discuss the Dasher interface and our adaption of it into virtual reality. The Dasher interface consists of many colored boxes, as can be seen in Figure 2. In its desktop variant the user can interact with this interface with a mouse or other devices. We opted for a direct port of the rendering with an input system similar to the desktop variant. For this, we used a raycast style interaction to replace the mouse and integrated a live rendering of the Dasher interface into our virtual environment, see Figure 1.
To input text in Dasher, the user can point towards a character among all characters of the alphabet on the right side, see Figure 2(a). The crosshair in the middle of the view is then shifted in the direction where the user is pointing and a "zooming" movement is performed. Zooming closer to the first column of characters, a second column of characters is shown recursively inside of each character in the first column, see Figure 2(b). Once the crosshair enters one of the boxes, the corresponding character is entered. Due to the continuously moving view, momentary inaccuracies can be corrected later on, similar to driving with a car on an empty road [10].
To assist in writing, the Dasher system is backed by a language model in the background. The language model "Prediction by Partial Matching" (PPM) with context length 5 and the escape strategy D (PPM5D), described in [28], was chosen as it compresses most English text to around 2 bits/character and is fast and efficient to evaluate. While in the Figure 2(a)-2(b) each letter is rendered the same size for explanatory reasons, Dasher normally renders characters bigger that are more probable to be the next character in the input string. Figure 2(c) shows an example with predictions from the PPMD5 model, where the user already input "us" and the language model predicts "use" and one layer deeper "used", "user" and "uses". PPM5D was slightly altered by the original authors to never assign a character a probability of 0, so every character can always be entered, even if it is highly improbable. A recording of the interface in action can be seen in this video[3].

Overall, the Dasher system allows inputting text in a continuous gesture, independent of the length of the alphabet used and allows for easy error correction without any mental context switch from inputting to auto-completion.
As one of the original authors, David MacKay, showed in one of his presentations [10], that the system can be controlled by nearly any input metaphor that can provide either a floating point number or even a two button interface. While the example input metaphors in his talk will not lead to the fastest entry speeds, they allow for great flexibility in the used input device.
To allow for text input in modern game engines, the license of the original code had to be adapted, which was done by the organization currently stewarding the project, the AceCentre[4], to enable our developments. We then modernized the original code base and prepared it for inclusion in game engines. We developed a plugin for the Unreal Engine which serves as the base for future developments and was used in our evaluation.

[3]https://www.youtube.com/watch?v=nr3s4613DX8
[4]https://acecentre.org.uk

4 EVALUATION

We believe that Dasher is a versatile text input method for immersive virtual environments due to its continuous gesture-based input paradigm as well as its flexibility in input device selection, which offers many opportunities for accessibility adjustments when required. We conducted a user study in order to get a first impression of the usability of the Dasher system in VR.
For the experiment, we rendered the Dasher interface onto a whiteboard in a virtual seminar room and used a 6-DoF controller with raycasting as a direct mouse replacement as described in Section 3.

4.1 Procedure

In our experiment, we asked participants to type a set of sentences that we selected randomly from the "Mixed-typical"-corpus of the Wortschatz Leipzig project [9]. However, in the random selection, we discarded sentences that contained religious or strong political content to not distract participants from the actual task. The set of sentences contained 19 sentences, leading to about $15-20$ min of text entry. To setup the Dasher system, we trained the language model on 30.000 other sentences from the same corpus and chose the German alphabet definition shipped with Dasher together with a colorblind color scheme. We chose to eliminate punctuation from the alphabet completely (erasing these symbols: ?!,.' and ¶), since those were not needed for the chosen sentences. Dasher by default enables a feature that adapts the moving speed continuously. While this is a good feature for expert users, it was considered confusing in pilot tests and we decided to turn it off.
We measured the input speed, error rate and user experience and additionally noted direct user feedback. To evaluate the user experience, we employed the well-established User Experience Questionnaire (UEQ) [15], which we evaluated with the provided evaluation spreadsheet.
In the preliminary tests and implementation phase, we noticed the high visual flow that is rendered by the Dasher system. As we were curious if this influences cybersickness, we aimed to evaluate this effect as well. We opted for a simple evaluation with a single question discomfort rating, relative to discomfort levels when arriving at the laboratory [21]. The participants were asked after the procedure to give a rating between 0 (same as coming in) and 10 (immediately want to stop).
In the beginning of the experiment, every participant got an introduction to the method and was granted a training period to their liking.
As a test system, we used a workstation with an Intel Core i9-10900X @ 3.7GHz and an NVIDIA RTX3090 in conjunction with an HTC Vive Pro 2.

4.2 Participants

We performed the study with 20 voluntary participants that we acquired via mailing lists. From these participants, 18 identified themselves as male and two as female. The average age was 25.55 years with a standard deviation of 5.78 years. Except for one participant, everyone had previous VR experience, 8 participants even regularly. Three of the participants had used Dasher at least once before.

4.3 Results

	Mean	SD	Rating
Attractiveness	1.192	1.14	⬆
Perspicuity	0.813	1.15	⬆
Efficiency	0.700	0.94	⇨
Dependability	0.913	1.17	⬆
Stimulation	1.888	0.69	⬆
Novelty	2.313	0.69	⬆

Table 1: Single scales of the User Experience Questionnaire [15], where each value is within [-3,3] and marked with green ($\geq$ 0.8), red ($\leq$ −0.8) and yellow for all values in between, based on the evaluation sheet of the questionnaire.

The results of the user experience evaluation can be seen in Table 1. The resulting scores for the different scales (Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation and Novelty) range from -3 to 3, where high scores ($\geq$ 0.8) are represented by a green arrow and low values ($\leq$ −0.8) are represented by a red arrow; yellow is used for intermediate values.
The verbal feedback we got from participants was mostly positive and some participants noted that they would like to use Dasher more often in virtual reality applications.
For the discomfort levels DasherVR scored relatively low (Mean 1.70, SD 1.89), but we observed some larger variations on an individual level: While most of the participants did not state anything about cybersickness, two of them, answering with 6 and 7 for the discomfort question, stated that they felt rather nauseous afterwards. We seek to improve on this in future research.

	min	max	mean	sd	median
WPM	1.54	22.56	9.4	4.03	8.7
NCER	0.0	28.57	0.92	2.85	0.0

Table 2: Words per minute (WPM) and Not Corrected Error Rate (NCER) for the DasherVR input method over all trials with their respective minimum, maximum, mean, standard deviation and median values.

In addition to the user experience, we measured text input speed (in words per minute, see [5]) and error rate (as Not Corrected Error Rate, see [25, 26]) as well. For the calculation of the error rate, we assume here that a character is counted as "deleted" only if the user has moved the cursor to the left side of the crosshair (zooming out) when the character is removed. This differentiation is important here, as moving the crosshair through other boxes while typing is part of the input metaphor and thus should not be counted towards an error rate. Descriptive statistics of the measured speeds and error rates can be seen in Table 2.
Based on a survey of research results on text input techniques [7], we see that the input speed, with an average of 9.4 WPM, is still behind the performance of physical keyboards. However, it appears to be roughly on par with other virtual text input methods, which motivates us to continue researching on how to improve DasherVR further. Moreover, we could see an improvement of the input speed as the experiment went on, which is visible if we only consider the last 5 sentences for the calculation of the speed. In this case the average input speed increases to about 12.56 WPM, outperforming many other virtual text input techniques listed in [7]. Additionally, we noticed that some sentences were relatively slow to enter, as the language model had low prediction scores for some characters/words. Post-hoc analysis of the average information content [24] (where higher means worse prediction) of each sentence, revealed a strong negative correlation between the information content and the input speed, $r = -.806, p < .001$, meaning that the prediction quality of the language model strongly influences the input speed.
The measured error rate is 0.92 NCER, which is relatively low and comparable to the techniques listed by [7].

5 CONCLUSION & FUTURE WORK

Based on our quantitative results and the feedback gathered from the participants, we believe that DasherVR is a promising first step towards fluent text entry in virtual reality. It also offers high customizability regarding the input metaphor to control the virtual cursor, which we deem highly beneficial in terms of creating accessible user interfaces. We are glad to have worked with the current stewards of the framework to modernize it and establish a shared codebase between the main project and DasherVR to contribute back to the main project as much as possible.
In the future, we plan to further test different input devices (analog sticks, touchpad and eye-tracking) and different rendering styles to make better use of the depth dimension offered by virtual reality. Additionally, we want to further tweak the Dasher rendering to lower the potential for cybersickness and increase general user experience. As we noticed the strong influence of the language model in our experiment, we started to cooperate with a project [23] that provides many language model implementations to potentially modernize the one used by Dasher.
According to the original Dasher authors, significantly higher input speeds of up to 34 WPM [31] with desktop Dasher and 25 – 30 WPM with an eye-tracker version of the system [32] were reached with even longer training. We could (so far) not replicate such high input speeds in virtual reality, but are keen to perform a longer term study with a further developed version of DasherVR to record a learning curve and improve on the measured speeds.

ACKNOWLEDGMENTS

The authors want to thank Sebastian Pick, Benjamin Weyers and Andrea Bönsch for their help in designing the study and Martin Bellgardt and Jonathan Ehret for help in prior testing and the evaluation of the statistical aspects. A very special thanks goes to the community and maintainers of the Dasher project and in addition to Will Wade and Gavin Henderson, working at the AceCentre, for their big efforts to change the license of Dasher for this work. Additionally, we want to thank all the study participants for their time and effort.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871260.

REFERENCES

[1] Jiban Adhikary and Keith Vertanen. 2021. Text Entry in Virtual Environments Using Speech and a Midair Keyboard. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2648–2658. https://doi.org/10.1109/TVCG.2021.3067776

[2] Jiban Adhikary and Keith Vertanen. 2021. Typing on Midair Virtual Keyboards: Exploring Visual Designs and Interaction Styles. In *IFIP Conference on Human-Computer Interaction*. 132–151.

[3] Costas Boletsis and Stian Kongsvik. 2019. Text Input in Virtual Reality: A Preliminary Evaluation of the Drum-Like VR Keyboard. *Technologies* 7, 2 (2019), 31.

[4] Doug A. Bowman, Christopher J. Rhoton, and Marcio S. Pinho. 2002. Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46, 26 (2002), 2154–2158. https://doi.org/10.1177/154193120204602611

[5] Steven J. Castellucci and I. Scott MacKenzie. 2011. Gathering Text Entry Metrics on Android Devices. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. 1507–1512. https://doi.org/10.1145/1979742.1979799

[6] Nell Chitty. 2013. User Fatigue and Eye Controlled Technology. (2013).

[7] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *Human-Computer Interaction. Recognition and Interaction Technologies (Lecture Notes in Computer Science)*, Masaaki Kurosu (Ed.). 419–437. https://doi.org/10.1007/978-3-030-22643-5_33

[8] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. 2020. PinchType: Text Entry for Virtual and Augmented Reality Using Comfortable Thumb to Fingertip Pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. 1–7. https://doi.org/10.1145/3334480.3382888

[9] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. 759–765.

[10] Google TechTalks. 2007. Dasher: Information-Efficient Text Entry. https://www.youtube.com/watch?v=wpOxbesRNBc

[11] Zhenyi He, Christof Lutteroth, and Ken Perlin. 2022. TapGazer: Text Entry with Finger Tapping and Gaze-directed Word Selection. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*. 1–16. https://doi.org/10.1145/3491102.3501838

[12] Lode Hoste, Bruno Dumas, and Beat Signer. 2012. SpeeG: A Multimodal Speech- and Gesture-based Text Input Solution. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. 156–163. https://doi.org/10.1145/2254556.2254585

[13] Lode Hoste and Beat Signer. 2013. SpeeG2: A Speech- and Gesture-based Interface for Efficient Controller-free Text Entry. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction (ICMI '13)*. 213–220. https://doi.org/10.1145/2522848.2522861

[14] Youngwon R. Kim and Gerard J. Kim. 2016. HoVR-Type: Smartphone as a Typing Interface in VR Using Hovering. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (VRST '16)*. 333–334. https://doi.org/10.1145/2993369.2996330

[15] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *USAB 2008*, Vol. 5298. 63–76. https://doi.org/10.1007/978-3-540-89350-9_6

[16] Jiaye Leng, Lili Wang, Xiaolong Liu, Xuehuai Shi, and Miao Wang. 2022. Efficient Flower Text Entry in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–11. https://doi.org/10.1109/TVCG.2022.3203101

[17] Xueshi Lu, Difeng Yu, Hai-Ning Liang, Wenge Xu, Yuzheng Chen, Xiang Li, and Khalad Hasan. 2020. Exploration of Hands-free Text Entry Techniques For Virtual Reality. *Proceedings - 2020 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2020* (2020), 344–349. arXiv:2010.03247 http://arxiv.org/abs/2010.03247

[18] Anh Nguyen, Samuel Bittman, and Markus Zank. 2020. Text Input Methods in Virtual Reality Using Radial Layouts. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. https://doi.org/10.1145/3385956.3422114

[19] Sebastian Pick, Andrew S. Puika, and Torsten W. Kuhlen. 2016. SWIFTER: Design and Evaluation of a Speech-based Text Input Metaphor for Immersive Virtual Environments. In *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*. 109–112. https://doi.org/10.1109/3DUI.2016.7460039

[20] Vijay Rajanna, Murat Russel, Jeffrey Zhao, and Tracy Hammond. 2022. PressTapFlick: Exploring a Gaze and Foot-Based Multimodal Approach to Gaze Typing. *International Journal of Human-Computer Studies* 161, C (2022). https://doi.org/10.1016/j.ijhcs.2022.102787

[21] Lisa Rebenitsch and Charles Owen. 2014. Individual Variation in Susceptibility to Cybersickness. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. 309–317. https://doi.org/10.1145/2642918.2647394

[22] Emily Rickel, Kelly Harris, Erika Mandile, Anthony Pagliari, Jessyca L. Derby, and Barbara S. Chaparro. 2022. Typing in Mid Air: Assessing One- and TwoHanded Text Input Methods of the Microsoft HoloLens 2. In *Virtual, Augmented and Mixed Reality: Design and Development (Lecture Notes in Computer Science)*, Jessie Y. C. Chen and Gino Fragomeni (Eds.). 357–368. https://doi.org/10.1007/978-3-031-05939-1_24

[23] Brian Roark and Alexander Gutkin. 2022. Design Principles of an Open-Source Language Modeling Microservice Package for AAC Text-Entry Applications. In *Ninth Workshop on Speech and Language Processing for Assistive Technologies (SLPAT-2022)*. 1–16. https://aclanthology.org/2022.slpat-1.1

[24] Claude E. Shannon. 1951. Prediction and Entropy of Printed English. *Bell System Technical Journal* 30, 1 (1951), 50–64. https://doi.org/10.1002/j.1538-7305.1951.tb01366.x

[25] R. William Soukoreff and I. Scott MacKenzie. 2001. Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. 319–320. https://doi.org/10.1145/634067.634256

[26] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. 113–120. https://doi.org/10.1145/642611.642632

[27] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-Based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. 1–13. https://doi.org/10.1145/3173574.3174221

[28] William J. Teahan. 1995. Probability Estimation for PPM. In *In Proceedings NZCSRSC'95*. http://www.cs.waikato.ac.nz/wjt

[29] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash Away! Longitudinal Study of Fast Text Entry by Eye Gaze. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications (ETRA '08)*. 19–26. https://doi.org/10.1145/1344471.1344476

[30] Keith Vertanen and David J.C. MacKay. 2010. Speech Dasher: Fast Writing Using Speech and Gaze. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 595–598. https://doi.org/10.1145/1753326.1753415

[31] David J. Ward, Alan F. Blackwell, and David J.C. MacKay. 2000. Dasher - a Data Entry Interface Using Continuous Gestures and Language Models. In *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*. 129–138. https://doi.org/10.1145/354401.354427

[32] David J. Ward and David J. C. MacKay. 2002. Fast Hands-free Writing by Gaze Direction. *Nature* 418, 6900 (2002), 838–838. https://doi.org/10.1038/418838a arXiv:cs/0204030

[33] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 113:1–113:21. https://doi.org/10.1145/3130978

[34] Sebastian A. Wills and David J. C. MacKay. 2006. DASHER - An Efficient Writing System for Brain–Computer Interfaces? *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 2 (2006), 244–246. https://doi.org/10.1109/TNSRE.2006.875573

[35] Tian Yang, Powen Yao, and Michael Zyda. 2022. Flick Typing: A New VR Text Input System Based on Space Gestures. In *Virtual, Augmented and Mixed Reality: Design and Development (Lecture Notes in Computer Science)*, Jessie Y. C. Chen and Gino Fragomeni (Eds.). 379–392. https://doi.org/10.1007/978-3-031-05939-1_26

[36] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. 2018. PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics* 24, 11 (2018), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581

[37] Zigang Zhang, Minghui Sun, BoYu Gao, and Limin Wang. 2021. 2-Thumbs Typing: A Novel Bimanual Text Entry Method in Virtual Reality Environments. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 530–531. https://doi.org/10.1109/VRW52623.2021.00147

Received 23 February 2023; accepted 10 March 2023