Bimanual Word Gesture Keyboards for Mid-air Gestures

Garrett Benoit
Baylor University
Waco, TX, USA
garrett_benoit@baylor.edu

G. Michael Poor
Baylor University
Waco, TX, USA
michael_poor@baylor.edu

Alvin Jude
Ericsson Research
Santa Clara, CA, USA
alvinjude@acm.org

CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053137

Abstract

Mid-air hand gestural interaction has generally been researched as a pointing device. However, recent research has shown potential for text input with the use of word gesture keyboards (WGK), where these forms of interactions require the input system to identify when the gesture has started and when it has stopped. Previous research has had success where the same hand moved the cursor, and performed the activation gesture. In this paper we introduce bimanual interaction for gestural interaction to perform text input with WGK, where one hand moves the cursor while the other hand performs the activation. In our user studies, the bimanual method demonstrated significantly higher results than the state-of-the-art single handed method. We achieved 16 words per minute; about 39% higher than the benchmark, and with significantly lower error rates.

Author Keywords

Gestural Interaction; Word Gesture Keyboards; WGK.

ACM Classification Keywords

H.5.2. User Interfaces: Input devices and technologies; Interaction styles (e.g., commands, menus, forms, direct manipulation)

Introduction

Touchless mid-air gestural interactions have been proposed for post-WIMP UIs [19] and, in recent years, have gained traction in part due to the availability of low-cost commodity hardware such as the Myo armband, Leap Motion and the XBox Kinect. There are 2 main methods in which gestural input is performed: glove-based or freehanded [5]. These forms of interactions have been shown to be useful as a pointing device on the desktop [14], 10-foot interaction with a distant display [20], and in 3D navigation such as maps [4]. Freehanded interaction have a number of benefits, such as the ability to be used in sterile environment [21] and seamless integration with public displays with lower health risks [22].

Gestural interaction used as a pointing device is generally able to perform 3 tasks: navigation, selection, and manipulation. In a 2-D output such as a computer screen, while navigation is generally done by moving a cursor on-screen [23]. Selection and manipulation on the other hand have been done in a few ways, including a grab or pinch gesture [15] with the same hand. Our work moves away from gestures for pointing, and instead aimed to investigate and improve existing knowledge in the area of gestural interaction for text or character input, which appears to be less popular compared to the same for point-and-select tasks.

We chose to use Word Gesture Keyboards (WGKs) as a means of text input with gestures, since they are commonly used in touch devices and allow users to perform input continuously without lifting their finger off the touchscreen. WGKs have been included by default in Google's Android device and have been shown to allow faster input compared to single character input methods. WGKs on mid-air gestures have likewise demonstrated higher performance tap-based single-character input. Both mid-air and touch-based interactions, have the same prerequisites; there needs to be a way to: (1) indicate a start and a stop per word, and (2) a way to continuously move a cursor. The current state-of-the-art method with mid-air gestures is Vulture [15], which uses a single-handed approach, where a 'pinch' gesture on the pointing hand provides the activation. The goal of our research was to investigate the viability of using bimanual interaction, where the non-dominant hand performs the activation gesture. Our experiments showed that bimanual interaction was better than single-handed mid-air interaction with higher WPM, and lower errors.

Related Works

WGKs – also known as shape-writing or continuous input – allow users to perform word entry on a touchscreen by tapping down on the first character, then swiping their finger onto the following characters in the word without lifting their finger [10, 26, 25, 24]. WGK was introduced on touchscreens as an alternative to tap-based input where users tap on each character at a time. WGKs have been shown to perform better than tap, where it demonstrated to allow for 25 words per minute (WPM) after 35 minutes of practice for beginners and up to 46 WPM for well-practiced phrases or expert-level experience [10, 28]. KeyScretch was a menu-augmented, word-gesture keyboard and was able to achieve text-entry rates of 44-50 WPM [6].

A traditional part of implementing word-gesture keyboards is to include shape-recognition algorithms in order to best determine what words the user was trying to create [10, 26, 27, 25, 24]. A word-recognition

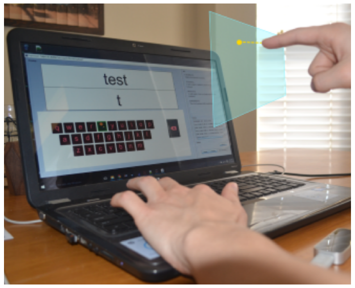

Figure 1 In the bimanual gesture, the word-gesture starts when the user presses on the spacebar. The user's finger is projected onto their pre-calibrated hyperplane along the Z-axis, which moves the on-screen cursor.

based approach has already been shown to work for word-gesture keyboards (e.g., SHARK2, Vulture) [10, 25, 15] and was therefore, outside the scope of this work. Instead, we used a pseudo-implementation without word-recognition, since it was known in advance what word-gestures a user was going to attempt to produce.

Schwaller et al. used two-handed interactions for manipulation of objects on-screen, where one hand performs pointing, while the other performs a "natural" activation gesture in the form of a "grasp" movement)[16]. The bimanual interaction in FlowMouse [23] was activated by the non-dominant hand using a laptop's mouse key. The interaction within FlowMouse was design to emulate an experience closer a track pad, limiting the potential power of the gesture interaction. These experiments demonstrate the capability of bimanual interaction, but not WGKs specifically. The goal of our research was to investigate if the findings from previous works were applicable to WGKs and mid-air gestures.

Input & Interaction Methods

The primary goal of this experiment was to assess the differences between bimanual and single-handed interactions. For this experiment the Leap Motion was used which incorporates light- and vision-based sensors affording bare-handed mid-air interactions unlike previous gesture work such as Vulture [15]. However, the Leap Motion was also more suited for desktop-based interactions causing us to ask participants to perform the interaction from a sitting position with their elbow rested on the table [2]. This method improves performance compared to an unrested position when users can calibrate their interaction space [9, 7].

Additionally, 2 touch-based interactions were used as benchmarks to contextualize the performance of the mid-air gestural interactions. In this way, we could hypothesize the upper limits, or the best possible performance that could be achieved with WGKs.

Bimanual Gesture

The bimanual interaction was designed to utilize two inputs: the Leap Motion controller and a standard keyboard and based on previous work. The Leap Motion tracked the pointer finger of either hand by projecting a quadrilateral plane in the air and snapping the movements of the pointer finger to the plane [9] as shown in figure 1. A "touch" was simulated using the secondary input [23], ie. a keyboard's space bar [2].

Single-handed Gesture

The essence of this interaction is that the 2 main interactions of the keyboard is performed with the same hand with the interaction design closely matching the current state-of-the-art: Vulture [15]. A pinch gesture indicated a start and a release of the pinch indicated a stop, as demonstrated in figure 2. We used the Leap Motion's implementation of the pinch via API.

Touch Screen

The Touch Screen Keyboard was used on a large tabletop touch. A large tabletop computer was used to ensure the interaction space was similar in size to the other gestural-interaction based input (14 × 10 cm, with key approximately 0.92 × 0.92 cm). This could not be accomplished in a mobile-phone where the interaction space is significantly smaller. This interaction was included as a benchmark as we believe this is the gold standard for WGK comparison. Participants used their finger to interact with the virtual

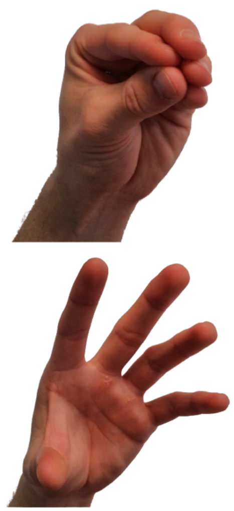

Figure 2 The single-handed word-gesture starts with the pinch (top) and stops with an open-palm (bottom)

keyboard on the screen in the same way as typical touch devices. Touch was detected when a participant's finger was pressed against the screen, and release was simulated when the finger was lifted from the surface.

Gesture + Surface
While touchscreen interaction is understood to be the best implementation of a word-gesture keyboard, the implementation has a significant difference in comparison to the gesture-based interaction. First, the actual input device used is different, therefore the pros and cons of the touchscreen vs the Leap Motion exists as a confounding variable. Second, the availability of a form of haptic input and immediate physical feedback; in the gesture based interactions, the user indirectly moves a cursor while in the touch based system, the user directly touches the characters. As a result, we introduced the gesture + surface interaction where the Leap Motion was used to detect touch and releases, and the keyboard was printed out on a piece of paper to inform users of the position of the keys. A stylus was used instead of the finger for better recognition by the Leap Motion, as shown in figure 3.

Methods
A user study was conducted with 18 participants (F=5) between 18 to 24, with a median age of 21. One participant self-reported as ambidextrous and opted to use their right hand, while another self-reported as left-handed and opted to use their left hand. All others self-reported as right-handed and chose to use their right. All participants reported previous experience with touch devices; 10 reported experience with word-gesture keyboards. None reported relevant impairment.

Experiment Design
Participants performed 15 unique trials per interaction, where each interaction consisted of entering one pre-selected word. Each word was 3 to 6 characters in length and selected based on an algorithm that produced the most dissimilar words in the Oxford English Dictionary based on keyboard position. The order in which participants used each interaction was determined with a balanced Latin-squared design.

For each trial, a word was chosen at random from the active keyboard's previously constructed dictionary and displayed on the screen. A blank text-area was positioned directly below the displayed word for the participant's transcribed text. Beneath both text areas, the keyboard interaction styles were virtually represented. The participants were then required to use the currently active keyboard interaction style to enter the displayed word using word-gesturing. Feedback was given in the form of a cursor moved on-screen, as well changing the color of the key selected. Participants were shown real-time word updates during the word-gesturing process, and were required to use the active keyboard interaction's backspace key to remove errors. However, already correct transcribed characters were protected from being deleted. Once a word was correctly entered, the participants released the simulated touch to move to the next word.

The researcher demonstrated the use of the input device by forming the word "test" with every interaction method. The participants are then given the opportunity to test the interaction until they felt they were able to efficiently and comfortably type each word with minimal errors. They were also informed that recalibration would be possible if required. At the end

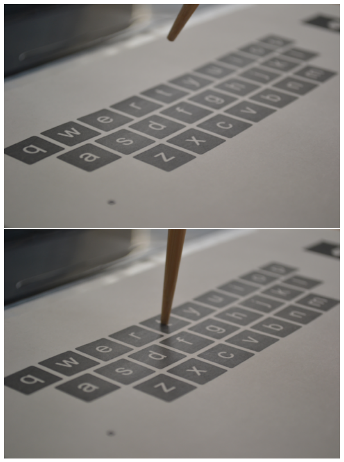

Figure 3 In the gesture + surface method, word-gesture start was simulated when the stylus hits the paper surface.

of the experiment, participants completed a survey to rank each keyboard from most to least preferred.

Keyboard & Test Words

Instead of building a fully featured word-gesture keyboard, a known-words implementation was used where the software has preexisting knowledge of the intended word. This was done with the intention to remove any confounding and/or usability factors in the design and implementation of a fully-functional version. The known-words implementation was created by analyzing the user's generated gesture and compared it to the expected gesture as it was being drawn, to determine which keys were being 'pressed.' The assumptions for detecting a key press were based on the word being gestured, and deviations made in the gesture's direction. To reduce the chance of erroneous keys being pressed, the sizes of the characters expected were exaggerated and the deviation threshold in gestures were lowered between key paths.

An interpolated trail with points at a minimum of 16 pixels apart was used in determining deviations in word-gesturing. The angle of detecting a deviation was 165° for all areas that were not on the expected path to the next key and was 90° while on the expected path. Deviations in gesture path had to be at least 48 pixels away from each other to be counted as a 'press'. The expected path between two keys comprised an area from the previous expected key to the next expected key with a width 62.5% larger than key size.

Evaluation Metrics

Performance of all WGKs were evaluated from various measures described below. Text-entry rates were calculated using the standard Words Per Minute (WPM) text gesture formula [13], shown in equation 1. We measured hand velocity as a metric as it was shown to be the single best predictor of performance with mid-air WGKs [15]. We evaluated errors in a few ways, based on existing techniques. Minimum Word Distance (MWD), based on Minimum String Distance [17, 12] and shown in equation 2. Correctness of a single word-gesture was measured with the Fréchet Distance (FD) between the expected word-gesture path and the participant's generated word-gesture path [8]. The Fréchet Distance between two curves P and Q, or in this case gesture-shapes, was defined as the minimum length leash needed to walk a dog when the person walks along P and the dog along Q. The two time-based measures reported were reaction time to simulate the start gesture, and the mean completion time per word.

Results

The exact figures for each metric measured during the experiment can be found in Table 1. A paired t-test between bimanual and single-handed showed statistical significance ($P<.001$) for all metrics. More important, there were high practical significance found in all metrics, as shown in the effect sizes (ES) measured in Cohen's d with pooled variance. ES above 0.2 is generally considered small, 0.5 is medium but visible to the naked eye, and 0.8 is considered large [3, 18, 11].

As a whole, we found that bimanual interaction allows word completion at about 32% faster, with about 39% more words per minute, and with significantly lower errors compared to the single-handed approach. It is also noteworthy that our single handed implementation with pinch at 11.3 was very close to that of Vulture (11.7 WPM), despite the inherent differences in both input devices.

$$WPM = \frac{|T|}{S} \times 60 \times \frac{1}{5}$$

Equation 1 Words Per Minute, where |T | is the length of the transcribed string and S is the amount of time (in seconds) taken to transcribe the word, from the time the first character was produced.

$$MWD(\%) = \frac{error\ words}{total\ words} \times 100$$

Equation 2 Minimum Word Distance. In our implementation, any word that was not correct on the first attempt was considered an error word.

The preference ranking of all of the keyboards reflected the trends in the other dependent measures. The TS was the most preferred, followed by SG, 2H, and 1H.

Metric		2H	1H	TS	SG
WPM	Mean	15.8	11.3	19.5	17.1
	stdev	2.2	2.0	3.0	3.3
	ES		2.14	-1.41	-0.46
Velocity	Mean	15.2	10.5	35.7	15.0
	stdev	5.1	3.2	6.7	3.9
	ES		1.10	-3.44	0.04
MWD	Mean	20.4	48.9	13.0	18.9
	stdev	16.9	25.2	13.4	14.0
	ES		-1.33	0.49	0.10
Fréchet Dist.	Mean	396.5	501.2	231.2	242.8
	stdev	102.1	42.8	26.7	30.4
	ES		-1.21	2.22	2.04
Total Dist.	Mean	1627.7	2165.3	1087.5	1423.9
	stdev	321.3	540.5	85.9	246
	ES		-1.21	2.30	0.71
React Time	Mean	1.41	1.71	1.24	1.22
	stdev	0.23	0.37	0.21	0.40
	ES		-0.97	0.77	0.58
Total Time	Mean	1627.7	2165.3	1087.5	1423.0
	stdev	321.3	540.5	85.9	246.0
	ES		-1.21	2.30	0.72

Table 1 A comparison of Bimanual (2H), Single-handed (1H), touchscreen (TS), and Surface Gesture (SG) with means, standard deviation, and effect size (ES) measured in Cohen's d

Conclusion, Discussion & Future Work

In this paper, we evaluated the efficacy of bimanual interaction for touchless mid-air word gesture keyboards, where the non-dominant hand was used for activation to start and stop the word-gesture. We demonstrated that it was better than then state-of-the-art technique, which was single-handed interaction, where a pinch gesture was used to start the word-gesture. We showed that bimanual interactions demonstrated better performance measured in words-per-minute, had lower error rates, and was better preferred by our participants in comparison to unimodal interaction. This demonstrated potential for future research in this area.

Overall, we see that TS dominates this interaction in all metrics, with SG close behind. This tells us that despite the improvements of bimanual over unimodal interaction, there is some space for improvements. Future research should consider these differences when designing interactions. For example, a haptic-based interaction should be considered if possible. It is noteworthy that there is very little difference between the bimanual and SG in terms of velocity, which has been noted to be the best predictor of performance. The performance levels of SG is what we will aim for in future iterations. We notice that the unimodal interaction had a much longer time to simulate the start gesture, as well as the total time per word.

Although the bimanual interaction performed better than unimodal in WPM, more work needs done to understand *why* the interactions were faster, the exact features that make it better, as well as its limitations. human factors such as fatigue and cognitive load could influence performance, and will therefore build experiments to test that hypothesis. Future work will also investigate if the differences are due to the input device as vision-based is generally considered more error-prone than glove-based input. We will also evaluate performance beyond the single pass, with more detailed training, varying number of characters per word, and with more difficult words.

References

1. 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 (UIST '12)*. ACM, 137–146.
2. Michelle A Brown, Wolfgang Stuerzlinger, and E. J. Mendonca Filho. 2014. The performance of un-instrumented in-air pointing. In *Proceedings of Graphics Interface 2014*. Canadian Information Processing Society, 59–66.
3. Robert Coe. 2002. It's the effect size, stupid: What effect size is and why it is important. *Paper presented at the British Educational Research Association annual conference, Exeter, UK* (2002).
4. Rita Francese, Ignazio Passero, and Genoveffa Tortora. 2012. Wiimote and Kinect: Gestural User Interfaces Add a Natural Third Dimension to HCI. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 116–123.
5. Dustin Freeman, Ramadevi Vennelakanti, and Sriganesh Madhvanath. 2012. Freehand pose-based Gestural Interaction: Studies and implications for interface design. In *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*. IEEE, 1–6.
6. Vittorio Fuccella, Mattia De Rosa, and Gennaro Costagliola. 2014. Novice and Expert Performance of KeyScretch: A Gesture-Based Text Entry Method for Touch-Screens. *IEEE Transactions on Human-Machine Systems* 44, 4 (2014), 511–523.
7. Darren Guinness, Alvin Jude, G Michael Poor, and Ashley Dover. 2015. Models for Rested Touchless Gestural Interaction. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. ACM, 34–43.
8. Sariel Har-Peled and Benjamin Raichel. 2014. The Fréchet Distance Revisited and Extended. *ACM Trans. Algorithms* 10, 1, Article 3 (Jan. 2014), 22 pages.
9. Alvin Jude, G Michael Poor, and Darren Guinness. 2014. Personal space: user defined gesture space for GUI interaction. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1615–1620.
10. Per-Ola Kristensson. 2007. *Discrete and Continuous Shape Writing for Text Entry and Control*. Doctoral dissertation. LinkÃű ping University, Sweden.
11. Chien-Ho Janice Lin, Katherine J Sullivan, Allan D Wu, Shailesh Kantak, and Carolee J Winstein. 2007. Effect of task practice order on motor skill learning in adults with Parkinson disease: a pilot study. *Physical therapy* 87, 9 (2007), 1120–1131.
12. 12. I. Scott MacKenzie and R. William Soukoreff. 2002. A Character-level Error Analysis Technique for Evaluating Text Entry Methods. In *Proceedings of the Second Nordic Conference on Human-computer Interaction (NordiCHI '02)*. ACM, New York, NY, USA, 243–246.
13. I. Scott MacKenzie and Kumiko Tanaka-Ishii. 2007. *Text Entry Systems: Mobility, Accessibility, Universality*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
14. Shahzad Malik and Joe Laszlo. 2004. Visual touchpad: a two-handed gestural input device. In *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 289–296.
15. Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-Air Word-Gesture Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 1073–1082.

16. Matthias Schwaller, Simon Brunner, and Denis Lalanne. 2013. Two handed mid-air gestural hci: Point+ command. In *International Conference on Human-Computer Interaction*. Springer, 388–397.
17. 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)*. ACM, New York, NY, USA, 113–120.
18. Katherine J Sullivan, Barbara J Knowlton, and Bruce H Dobkin. 2002. Step training with body weight support: effect of treadmill speed and practice paradigms on poststroke locomotor recovery. *Archives of physical medicine and rehabilitation* 83, 5 (2002), 683–691.
19. Andries Van Dam. 1997. Post-WIMP user interfaces. *Commun. ACM* 40, 2 (1997), 63–67.
20. Radu-Daniel Vatavu and Ionut-Alexandru Zaiti. 2014. Leap Gestures for TV: Insights from an Elicitation Study. In *Proceedings of the 2014 ACM International Conference on Interactive Experiences for TV and Online Video (TVX '14)*. ACM, New York, NY, USA, 131–138.
21. Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Craig Feied, Mark Smith, and Jon Handler. 2007. Gestix: a doctor-computer sterile gesture interface for dynamic environments. In *Soft Computing in Industrial Applications*. Springer, 30–39.
22. Robert Walter, Gilles Bailly, and Jörg Müller. 2013. StrikeAPose: revealing mid-air gestures on public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 841–850.
23. Andrew D Wilson and Edward Cutrell. 2005. FlowMouse: a computer vision-based pointing and gesture input device. In *IFIP Conference on Human-Computer Interaction*. Springer, 565–578.
24. Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104.
25. Shumin Zhai and Per-Ola Kristensson. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, 43–52.
26. Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (2012), 91–101.
27. Shumin Zhai, Per-Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the Iphone: From the Laboratory to the Real World. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, 2667–2670.
28. Shumin Zhai, Alison Sue, and Johnny Accot. 2002. Movement Model, Hits Distribution and Learning in Virtual Keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 17–24.