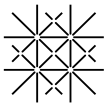


University of Basel

# Word-Gesture Typing in Virtual Realty

Bachelor thesis

Databases and Information Systems
Department of Mathematics and Computer Science
Databases and Information Systems
https://dbis.dmi.unibas.ch/

Examiner: Prof. Dr. Heiko Schuldt
Supervisor: Florian Spiess

Philipp Weber
phil.weber@stud.unibas.ch
19-051-697

20.06.2022

# Acknowledgments

So Long, and Thanks for All the Fish. And the template.

# Abstract

Text-entry is one of the most common forms of computer-human interaction and indispensable for many tasks such as word processing and some approaches to multimedia retrieval. The conventional keyboards everybody knows have long-established as the main text input method for desktop and laptop computers and even for touchscreen based devices they are very useful. But when it comes to virtual reality (VR) and augmented reality (AR), conventional keyboards might not be the best solution. With today's technology, VR and AR lack of tactile feedback and accurate finger tracking. As a result, text input for VR and AR is still an area of active research.

In recent years, word-gesture keyboards (also called slide-to-type keyboards) have been introduced in most major smartphone operating systems. Word-gesture keyboards look more or less like a conventional keyboard. But instead of tapping on the different keys, words are written with gestures. Now, the idea is, that they might also work well with VR/AR.

# Table of Contents

# 1 Evaluation

In this section we want to talk about the evaluation as a whole. We want to look at the phrases we took, how we carried it out, the results observed and compare it to other results.

## 1.1 MacKenzie Phrase Set

One precondition for the evaluation is to use the MacKenzie Phrase Set[1]. Basically, this is just a set of 500 phrases. According to the paper [4], such a phrase set should use phrases of moderate length, that are easy to remember and representative for the target language. These phrases do not contain any punctuation. Some of them use uppercase characters, but the authors mention, that participants can also be instructed to ignore the case of the characters.
Some statistics for the whole phrase set, also found in the original paper [4]: The MacKenzie phrase set consists of 500 phrases, that have a minimum length of 16, a maximum length of 43 and an average length of 28.61 characters. On the whole, 2712 words were used, which consist of 1163 unique words. A phrase consists of a minimum of 1, a maximum of 13 and on average of 4.46 words.

## 1.2 Task of the Participants

The task of the participants is to copy 15 "random" phrases. They are not really random, but adjacent phrases from the downloadable MacKenzie Phrase Set (http://www.yorku.ca/mack/PhraseSets.zip). As they are not in a specific order, e.g. alphabetic order, we decide to do it like this.
TODO: PICTURE OF EVALUATION SCENE. The participants see two text fields. On the top is the phrase to copy, on the bottom the words/phrase they write. If the given phrase matches the user inputted phrase, a sound sounds, such that the participants know when they finish one specific phrase. After that, a new phrase appears until 15 phrases are correctly inputted. If an incorrect word is entered, the user either can use the word

[1] http://www.yorku.ca/mack/PhraseSets.zip

suggestions (fig **??**) or delete the wrong word and try to write it again. If a mistake is only noticed later on, the participants have to remove all words and characters up to and including the wrong word by using the backspace button. After this first step, in the second step, we briefly explain two functions of the keyboard, which they can test afterwards. First the scaling buttons and then the function to add a new word. This is important, because we want to know if they find these functions useful and well implemented.
The last step of the evaluation is to fill a questionnaire. First it has some general questions about the participant's experience in VR. Then there is a block of twelve questions. The first ten are from the system usability scale and the last two are for the previously mentioned functions they test. Per question, there are five possibilities to set the cross. From 1 (strongly disagree) to 5 (strongly agree). The questions are structured in such a way, that if the user is highly satisfied with everything, they would alternately make a cross at the 5 and 1. TODO: questionnaire as appendix.

## 1.3 Carry-out

To carry out the evaluation, we used two different VR systems. One was a setup with a HTC vive and HTC vive controllers. The other one included an Oculus Rift headset with corresponding controllers. Even though these are two different systems, it did not change much for the participants. In fact, only the controllers and their buttons differ a bit.
To find participants, we wrote an email to students from our university, and asked family members and friends. All in all, eleven people got in touch with us and participated at our evaluation. Every participant got the same explanation to give everybody the same foundation of knowledge.
We told them that if they are close enough to the keyboard, then the color gets a bit brighter, and they are in the keyboard's hitbox. We said, the keyboard is movable, if they press and hold the controller's grip button in the hitbox of the keyboard. If they release it, the keyboard gets static again and stays where it got put.
To write, they do also have to be in the hitbox of the keyboard but not pressing and holding the grip button, but the trigger button. Then they have to make a gesture over the characters of the keyboard to write a word. We also told them, that if they do a full gesture and a word longer than one character is written, a space is automatically put behind the word. We also said to them, that single characters could be inputted by clicking on a key of the intended character. If they did so, no space is put, and they have to do it on their own. In the English language, this is particularly important for the words "I" and "a".
We also told them, that if they made a gesture and a word was written, there may be one to four other choosable words. They could pick from them, if the word written in the text field is the wrong one. We especially mentioned the word "the". All the time "thee" would be written as the best match, therefore they would have to correct it every time.
They were also informed about the backspace. So, that if they use the backspace button after writing a word, the whole word gets deleted and afterwards only single characters get deleted.
We also told the participants, that we have enough time and that they should not hurry,

but rather look, that the inputted words are correct. Because if they are not correct, they have to use the backspace a lot of times.
In total for the eleven participants, 165 phrases were used. The average word length was 4.16 with the minimum length of 1 and a maximum length of 12. In total, 856 words were written, and it consisted of 460 unique words. We also made sure before the evaluation, that all the words in these phrases are in our word lexicon, such that the participants could write every word with a gesture.

## 1.4 Results

Now, we want to talk about the results and some statistics we gained through the evaluation.

### 1.4.1 System Usability Scale

First, we begin with the results of the SUS questions. These ten questions were:
1. I think that I would like to use this system frequently when I work in VR
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I need the support of a technical person to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system

| Question | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 4 | 5 | 4 | 4 | 4 | 5 | 4 | 4 | 5 | 4 |
| 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 5 | 4 | 5 | 4 | 5 | 5 | 4 | 4 | 5 | 5 | 5 |
| 4 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 2 | 1 | 1 |
| 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 5 |
| 6 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 4 | 1 | 1 | 2 |
| 7 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 5 | 5 | 4 |
| 8 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 |
| 9 | 4 | 3 | 5 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 |
| 10 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| | | | | | | | | | | | |
| Score | 85 | 85 | 97.5 | 87.5 | 90 | 85 | 90 | 75 | 90 | 97.5 | 87.5 |
| | Excellent | | Good | | | | | | | | |

| |
|---|
| strongly agree for positive question, strongly disagree for negative questions |
| agree for positive question, disagree for negative questions |
| neutral |
| disagree for positive question, agree for negative questions |
| strongly disagree for positive question, strongly agree for negative questions |

Figure 1.1: System Usability Scale with all the given points per question from every participant

Question 9 got the worst score, but with it being a 4.09 out of 5, it is still pretty good. From these values, we can calculate a usability score. Every question with an even number is a negative one. That means, a score of one or “strongly disagree” is the highest possible. For the other questions, a score of five or “strongly agree” is the best possible score. So, from the odd numbered questions we have to subtract 1 from the average score. And for the even numbered questions we have to subtract their average score from 5. At the end, we have to sum up these ten newly calculated values and multiply them by 2.5. Our calculated usability score is 88.18. This is a high score, because from a score of 85.5 points, one talks about an excellent system usability. Therefore, we are really satisfied with the results of this.

We do also have two other questions about the scale and the “add word” function:

11. The function to add words is well implemented and easy to use

12. The function to scale the keyboard is unnecessary

Question 11 got a score of 4.55 out of 5 and question 12 got a score of 2.18, whereby 1 would be ideal. We conclude from these two questions, that the “add word” function makes a good impression whereas the scale function does not perform so well.

### 1.4.2 Writing Speed

One important thing of our evaluation is to find out, how fast users can write with our word-gesture keyboard. As unit to measure these values, we take the "words per minute" wpm. We calculate the wpm with following formula:

$$WPM = \frac{\mid T \mid}{S} \times 60 \times \frac{1}{5} \tag{1.1}$$

where $T$ is all the phrases a participant had to write, hence $\mid T \mid$ is the number of characters a participant had to write. $S$ is the time in seconds they used to write all 15 phrases.

Table 1.1: average wpm, lowest wpm and highest wpm per participant. For the first three, we failed to get all data.

| participant | average WPM | lowest WPM | highest WPM |
|---|---|---|---|
| 1 | 11.457 | - | - |
| 2 | 12.19 | - | - |
| 3 | 13.055 | - | - |
| 4 | 11.609 | 5.3 | 25.5 |
| 5 | 12.578 | 6.83 | 21.65 |
| 6 | 10.285 | 5.27 | 19 |
| 7 | 12.423 | 6.1 | 24.41 |
| 8 | 16.056 | 8.28 | 30.74 |
| 9 | 13.363 | 7.96 | 24.15 |
| 10 | 17.118 | 7.98 | 24.45 |
| 11 | 10.067 | 4.71 | 14.55 |
| average | 12.75 | 6.55 | 23.06 |

In Table 1.1 you can see how fast in average the participants were able to write their 15 phrases. We do also list the lowest and highest value. Everything is measured in words per minute.

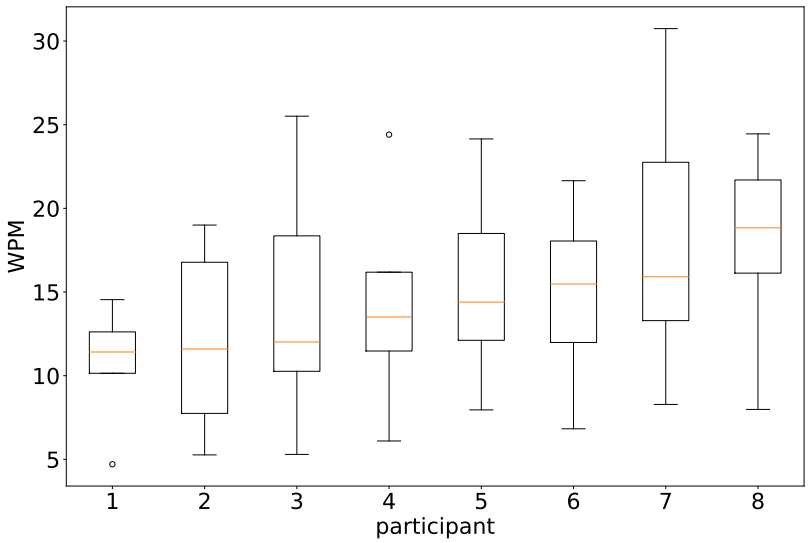



Figure 1.2: for participants 4-11: lowest wpm, 25% quantile, average wpm, 75% quantile and highest wpm

To understand the values of Table 1.1 a bit better, we make a so-called boxplot for every participant, for whom we have the data. We can see in fig 1.2, the higher the participant's median, most of the time they do also have higher lowest wpm value. The lowest wpm values mostly come about because a participant made a mistake and had to delete a lot and basically write the phrase two times. On the other hand, most of the highest WPM values come about because a participant made no mistake in writing the phrase. The rectangle in the middle of the two bars shows how consistent or inconsistent a participant's writing speed is. The lower bound is the 25% quantile, the upper bound the 75% quantile. This means, if the rectangle is less high, the writing speed is more consistent.
We cannot really find anything that combines writing speed and consistency by looking at our measurements.
As mentioned above, if the participants only recognize an error at the beginning of the phrase when they almost finished it, they have to use a lot of backspaces and more or less have to write the phrase a second time. These can pull down the average wpm a lot. Therefore, we calculate the average wpm for the participants 4-11, where we only look at "perfect" phrases, that were written without using a backspace. First of all, the average wpm for these eight participants over all phrases is 12.94. The average wpm of these "perfect" phrases only is 16.53, which is about 28% higher.

Next, we want to find out, if the writing speed of the users has something to do with their experience in, on one hand VR writing and on the other hand word-gesture keyboards.

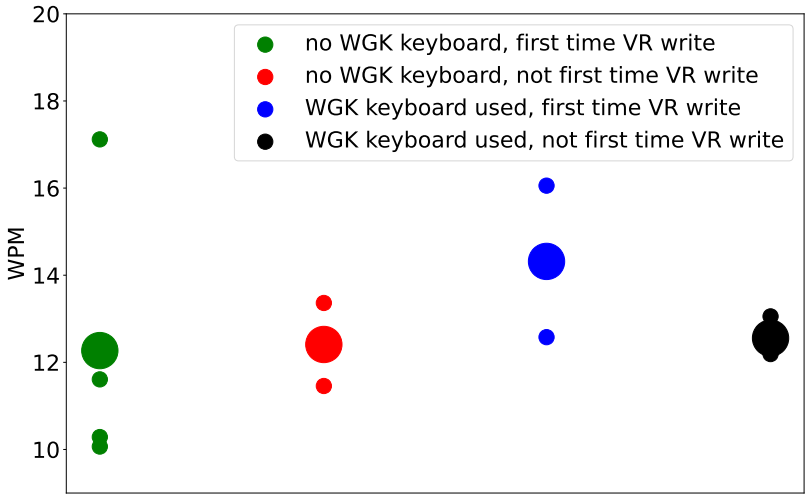



Figure 1.3: Each small dot shows the average wpm for a user. The big dots represent the average wpm per group. The colors are for the different groups of experience in VR writing and word-gesture keyboards

In Fig 1.3 we can see, that the prior knowledge, that some participants have, did not really help them to write faster. In fact, the fastest group was the one, that is experienced with word-gesture keyboards, but not with writing in VR. But we think due to the small group sizes this is not really presentative. Therefore, we can not conclude much about this, but it could have had a more interesting result.

Another thing we wanted to analyze is the writing speed compared to the age:

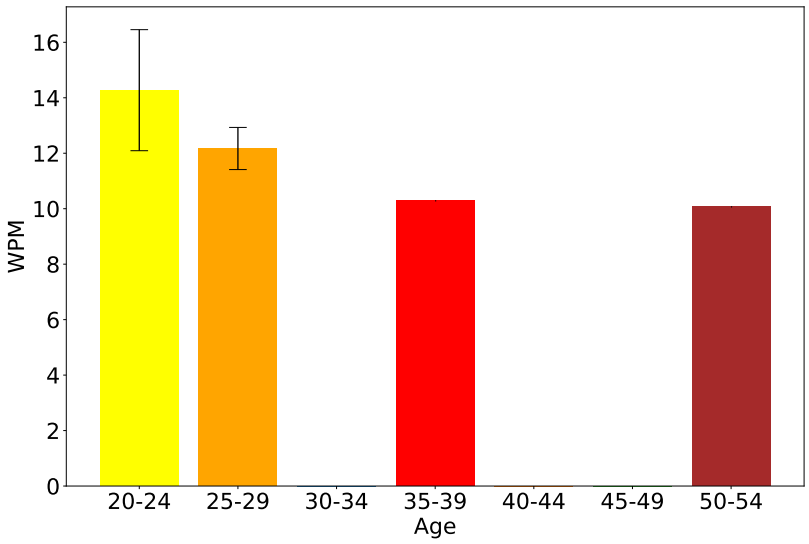



Figure 1.4: wpm values sorted by age groups of five years. The 20-24 and 25-29 groups also have a standard deviation error bar, because they contain of five and four participants. The 35-39 and 50-54 groups do not have an error bar, since each of them only contains of one participant

We can observe in Fig 1.4, that at least among our participants, the wpm value decreases with the increase in age. This means, the older participants were a bit slower than the younger ones.

### 1.4.3 Error Rate

In this section, we will look at different error rates. First of all, we want to find out, which words caused the most errors. Then we investigate some errors caused by the participants. At the end of this section, we take a look at the errors mostly caused by the system's condition.

#### 1.4.3.1 Most Frequent Error Words

We look into all the words that were not the best match, so all the words a participant either corrected or not and also the ones that were not even in the keyboard's word suggestions. The top ten such words are:

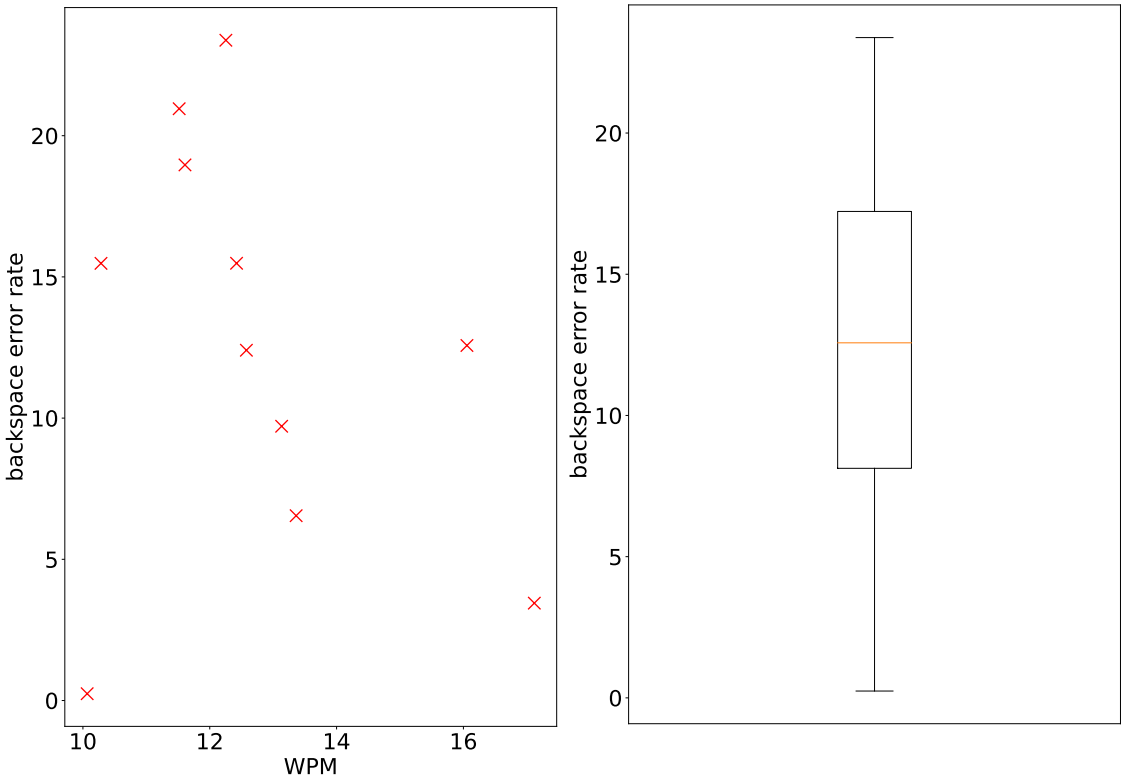
Table 1.2: most frequent error words

| word | times wrong |
|---|---|
| the | 53 |
| is | 17 |
| to | 14 |
| of | 12 |
| in | 9 |
| for | 8 |
| do | 5 |
| all | 3 |
| more | 3 |
| see | 2 |

As we can see in Table 1.2, the word that is responsible for most of the errors is “the”, which caused 53 of them. In this list, we can also see other words whose errors can be avoided by a better implementation. For example “ìn” and “more”. The best match, when a participant wanted to write these words, were “thee”, “inn” and “moore”, which are words, that do not appear that much in plain language. For other words like “to” or “of”, whose best matches were “too” or “off”, it is maybe not as easy to get the right word. Because all of them appear fairly often in plain language.

#### 1.4.3.2 Backspace Error Rate

The erroneous keystroke error rate (EKS ER) [1] measures the ratio of the total number of erroneous keystrokes to the length of the phrase that has to be written. For our calculations, we will take a similar formula but not exactly the one for the EKS ER. In another paper, Chen et al. [3] also used a variant of EKS ER. Instead of the erroneous keystrokes, they took the number of times the backspace key was used. We call this the “backspace error rate”.




(a) backspace error rate compared to wpm

(b) a boxplot for the backspace error rate over all participants. The orange line is for the median, the rectangle shows the 25% and 75% quartiles, and the upper and lower whiskers show the maximum and minimum.

Figure 1.5: shows the backspace error rate in two different ways. (a) with a comparison to the wpm of each participant and (b) a boxplot combining all participants' backspace error values.

In Figure 1.5(a) we show which wpm value was reached at which backspace error rate. We can not find much of a correlation. But one thing that can be seen is, that the participant with the lowest wpm seemed to be very careful not overlooking wrong words that could be corrected by choosing the right keyboard suggestion. The participant with the highest wpm has the second-lowest percentual usage of backspaces. Therefore, they seem to get used to the system and its suggestion words very fast and good. Overall, there seems to be a little trend, that participants with higher wpm values had to use fewer backspaces than participants with lower wpm values.
In Figure 1.5(a) we can see that the median value is around 12.5%. In fact, the average backspace error is 12.65%. This means, for every input of an amount of words with 100 correct characters about 13 backspaces were used.

#### 1.4.3.3 Total ER

The next thing we calculate is the Total ER according to Soukoreff and MacKenzie [5]. First of all, we have to clarify, that for this calculation we take another look attention the transcribed text than we did in the execution of the evaluation. In the Section 1.2 we mentioned that the phrase, a participant wrote, had to correspond exactly to the given phrase, otherwise it would not have gone to the next one. For the measure of the total ER value, we take the transcribed text as if a participant would not have been able to press the backspace. With only one exception, the backspace would have been allowed, if neither the best match nor the word suggestions had the right word and a wrong one is written. For the total ER value, we take following formula:

$$Total\ ER = \frac{INF + IF}{C + INF + IF} \times 100\% \tag{1.2}$$

Where $C$ is the number of correct characters in the transcribed text, $INF$ denotes the number of incorrect and not fixed characters in the transcribed text compared to the given phrases, identified with the minimun string distance and $IF$ denotes the keystrokes in the input stream, that are not in the transcribed text and not editing keys.
(However, we will not take these values exactly as stated above as Soukoreff and MacKenzie [5] did. For $C$ we take the number of correct characters of each 15 phrases a participant had to input. This means, we take the amount of characters that had to be written and subtract one for every character that is not in the previously defined transcribed text, but should be. One thing counting into this are missing spaces most of the time after words with one character as “I” or “a”. Also, some characters from not immediately corrected words can count into it.
Next, for $INF$ we do not count every character of an error word as error, but we calculate the minimum string distance (MSD) between the error word and the word it should be. The MSD value is the number of single character insertions, deletions or substitutions to get from the wrong word to the right one. For example, to get from “thee” to “the”, we need one deletion, therefore the MSD value would be 1.
Finally, for $IF$ we count the amount of backspaces a participant used to delete a wrong word, where the correct one was not even in the keyboard’s word suggestion.) The following table shows our calculated results:

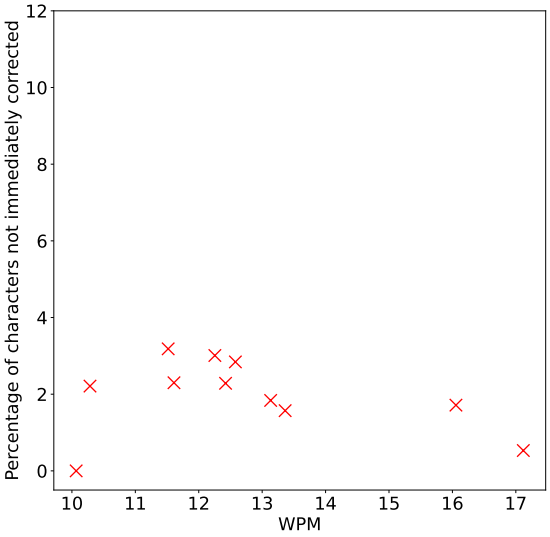
Table 1.3: Total ER per participant

| participant | total ER |
|---|---|
| 1 | 3.15% |
| 2 | 2.95% |
| 3 | 1.82% |
| 4 | 2.54% |
| 5 | 3.06% |
| 6 | 2.18% |
| 7 | 2.73% |
| 8 | 2.25% |
| 9 | 1.81% |
| 10 | 1.05% |
| 11 | 0% |
| average | 2.14% |

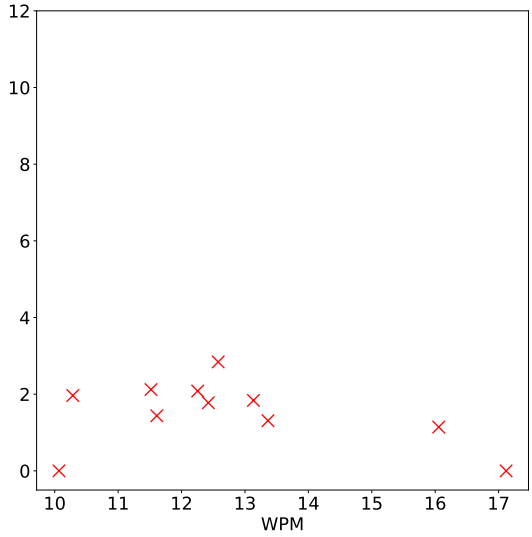
As we can see in Table 1.3, the calculated total ER values are quite low. The averge is only 2.14% with a standarddeviation of 0.91. Right now, we can not tell much about this, but later, we will compare these values with others from another paper.

#### 1.4.3.4 Participant Conscientiousness

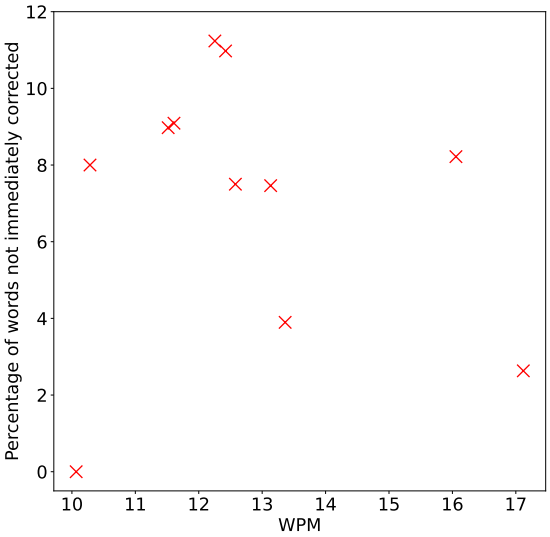
For the next calculation, we investigate the percentage of not immediately corrected words together with the wpm. Firstly, we will do this with characters and then with the whole words. For the calculation with the characters of an error word we take the MSD values as we did in the previous calculation with the total ER. Then we want to compare these two with the same approach, but without considering the errors happened because of “the”, “in” and “more”. We decide to remove these three words, because they could be avoided by a small change in the best match calculations.



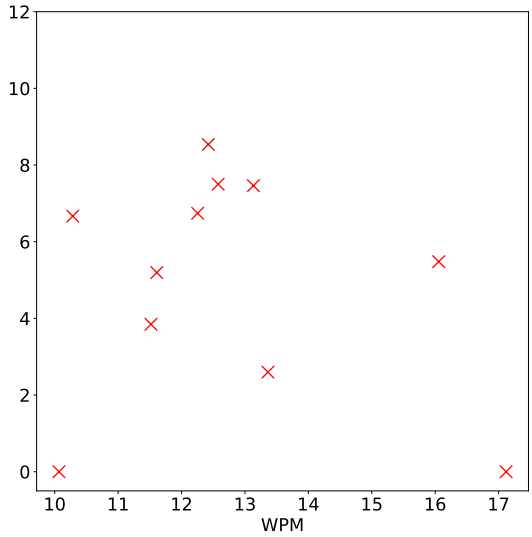

(a) Not corrected characters with wpm




(b) As (a), but without “the”, “in” and “more”




(c) Not corrected words with wpm




(d) As (c), but without “the”, “in” and “more”

Figure 1.6: (a) shows the percentage of characters calculated with the MSD from wrong words, that were not immediately corrected with the word suggestions. (b) is the same as (a) but errors from “the”, “in” and “more” are ignored. (c) is the same as (a) but not with characters, but with the words as whole. (d) is the same as (b) but not with characters, but with the words as a whole

In Fig. 1.6(a) and Fig. 1.6(b) we calculate the MSD of the words, a participant did not correct, although they had the chance to do it with the keyboard’s suggestions. On

the y-axis, we look at the percentage, that these characters make up in comparison to all characters the participant had to write in all 15 phrases. In Fig. 1.6(a) we look at all wrong words. But a lot of the errors came from the words “the”, “in” and “more”, that could have been prevented as mentioned before. Therefore, in Fig. 1.6(b) we look at the errors without considering these three words.
We can see that the percentage of characters that are not corrected lies between about 0-3.5%. Overall we can observe that the percentages go a bit down from Fig. 1.6(a) to Fig. 1.6(b). Looking at the wpm, we can say that the writing speed does not really affect the percentage of characters not immediately corrected and is to some extent the same for every participant.
In Fig. 1.6(c) and Fig. 1.6(d) we do the same, but without the MSD of the wrong words but just look at the amount of them.
For these two figures, we can observe that the percentages also drop if we do not consider the error of the three previously mentioned words. Here it makes a bigger difference than in Fig. 1.6(a) and Fig. 1.6(b). Therefore, it seems that these three words do not contribute much to the total MSD value. Another thing we can see, is that the percentage of not corrected words lies between 0% and about 12% which is much higher than the percentages for the characters. This is because we work with the MSD. If a word is wrong, most of the time, our keyboard will not write a totally wrong word, if the gesture is right to some extent. It will consist of similar letters as the right word. Therefore, not the whole word’s characters will be counted into this value.

To take another look at the participants’ attention, we want to calculate the value of participant conscientiousness. As for the total ER, we also take the formula for the participant conscientiousness value from the paper of Soukoreff and MacKenzie [5]:

$$Participant\ Conscientiousness = \frac{IF}{INF + IF} \tag{1.3}$$

Normally, they would take the values of $IF$ and $INF$ as mentioned in Section 1.4.3.3 (not as we took them, but as Soukoreff and MacKenzie [5] took them). For us, the problem is that we emanate from the scenario, where a participant is not able to use backspaces apart from the special case. Therefore, there will not be any characters in the input stream that are not in the transcribed text besides the special case words and the according backspaces. To get a better value for the participant’s conscientiousness we decide to work with words only, not characters. Thus, for $INF$ we count all the words not immediately corrected by a participant and for $IF$ the words that were corrected by using the keyboard’s suggestions. We think, that the calculation makes more sense this way when looking at our keyboard, that writes most of the time whole words and not single characters.
A score of 100% would be perfect. This would mean, that a participant corrects every word with the word suggestions, if it is possible. The average value we get here is 66.28%, which means about two third of the words, that can be corrected with the suggestions, are corrected this way. The worst value is 33.33% and the best value a participant achieved is 100%. In our opinion, this score is not too bad. A lot of the participants are not used to word-gesture-keyboards and do not work in or use VR that much. We think, if they

used the keyboard over a longer period of time, they would get used to the words, that are sometimes harder for the keyboard to get as best match, and would pay more attention to use the word suggestions then.
Interesting is, that if we let away the errors caused by “the”, “in” and “more”, the average value decreases to 63.73%. This is an indication of the fact, that the participant seemed to get used to the errors caused by these three words and fixed them by using the word suggestions.

#### 1.4.3.5 System Error Rate

Here we do not want to see if a participant was paying attention or not, but rather if our algorithm finds the right words most of the time. We count all the words that were neither the best match nor in the word suggestions, such that a participant had no chance to get this word right the first time. If this happened multiple times to a word, we count it in every time. We decide to name this error rate the system error rate, although a participant could have done a really bad gesture which then would not really be the system's fault to not find a word

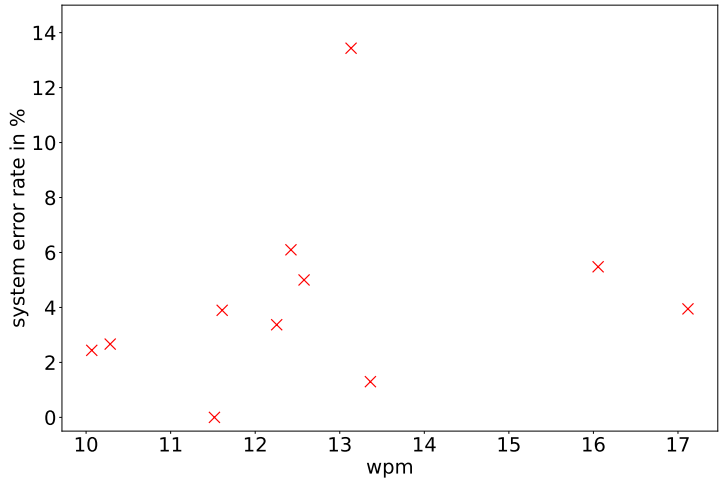



Figure 1.7: percentage of words that were not found by the system neither as best match nor as suggestion

In Figure 1.7 we can see what percentage of the words were not found by the system, neither directly as best match, nor as word suggestions, which a participant could choose from.
We can see, that the system error rate, does not have a correlation with the wpm. We can not see a tendency, that either the ones with high wpm or the ones with low wpm have fewer problems with not being able to write the right word. The average system error rate is 4.33%, which means that about every 23rd word made a problem.
On one hand, the average length of a word of the phrases we used is 4.16, on the other hand,

the average length of a word the system has not found in the first try is 7.03. This tells us, that words with a length way bigger than the average one, are more error-prone. We think this makes sense, because the longer the word, most of the time, the longer the gesture. In a bigger gesture, deviations from the perfect graph are harder to prevent than in a small gesture and the system might not output the right word.

### 1.4.4 Feedback

The full list including all verbatim feedback from every participant can be found in the appendix. Here, we just want to highlight the most frequently addressed points.
Some participants find the best matches and suggestions sometimes confusing. The most mentioned example is that "thee" is preferred over "the".
The visualization of spaces or the current position are another thing that is frequently addressed. Some participants find it unclear where the cursor is, hence if a space is missing or not. They suggest to implement some kind of visual indication, that indicates, where the cursor currently is. We can say about this, that this problem existed in the evaluation environment, but in vitrivr-VR, the program that our keyboard is manly developed for, has these kinds of cursor visualizations.
We also got a lot of praise and most of the time we saw a cheerful face when the VR headset got taken off. Some find it surprisingly intuitive, and others have a good feeling about using it.

## 1.5 Discussion

In this section we compare our results to other works' results and talk about some changes we did because of the feedback.

### 1.5.1 Result Comparison

First, we begin with the wpm. Boletsis and Kongsvik [2] evaluate in their paper four different VR input methods, a raycasting, drum-like, head-directed input and a split keyboard. The first one is a keyboard where a user can select letters by pointing a ray with a controller on it. For the second one the controllers simulate drum sticks in VR and letters have to be pressed by them. For the third keyboard a user has to aim with the head for the letters and press a button on the controller to input. The last keyboard is one, that is split into two halves, one assigned to each controller.
Another paper we want to compare the wpm with is from Chen et al. [3]. One of the keyboards they evaluate is a word-gesture keyboard where a user points with a ray from the controller to it and makes gestures like this.

In Table 1.4 we listed the results of Boletsis and Kongsvik [2], Chen et al. [3] and our measurement of the wpm value. Both had a similar approach to the evaluation as we did, with one difference. They did use the same ten phrases for every participant and keyboard type.
As we can see, our keyboard lines up in the middle. It is not the one with the lowest wpm,

Table 1.4: wpm for different VR text input methods

| text input method | wpm |
|---|---|
| Raycasting keyboard | 16.65 |
| Drum-like keyboard | 21.01 |
| Head-directed input keyboard | 10.83 |
| Split keyboard | 10.17 |
| Word-gesture raycast keyboard | 16.43 |
| Word-gesture keyboard | 12.75 |

but also not the one with the highest.

Another comparison we can do with the results of Boletsis and Kongsvik [2] is the comparison of the total ER value.

Table 1.5: total ER for different VR text input methods

| text input method | wpm |
|---|---|
| Raycasting keyboard | 11.05% |
| Drum-like keyboard | 12.11% |
| Head-directed input keyboard | 10.15% |
| Split keyboard | 8.11% |
| Word-gesture keyboard | 2.14% |

The total ER for our keyboard seems to be much better than for the other input methods. A simple explanation could be that a user can make an error with every input. For the first four keyboards in Table 1.5 one input is equal to one character. Thus, if a user wants to input a phrase, they need to make inputs equal to the number of characters the phrase consists of. With a word-gesture keyboard, a user only has to make inputs equal to the amount of words in the phrase, which is usually much lower. And if the word is not the correct one, most of the time, it will not differ a lot from the word, that is intended to write. Therefore, in this case, there will not a lot of characters be added to the $INF$ value for Equation (1.3).

The last comparison we want to make is with the EKS ER value from Chen et al. [3]. They state in the paper, that they used the EKS ER, but without the erroneous keystrokes, but the number of backspaces used. Therefore, this is defined the same way as our backspace error, and we can compare the values.

Table 1.6: total ER for different VR text input methods

| | |
|---|---|
| Word-gesture keyboard raycasting | 15.64% |
| Our word-gesture keyboard | 12.65% |

In Table 1.6 we can see, that the error values are pretty much the same, although our value is a bit better. A reason for this might be, that they possibly did not implement, that a word can be deleted as whole. For example, if a word is wrong, with our implementation the user can just press the backspace key once and delete the whole word (only the first one,

afterwards it deletes single characters).

Overall we think, that the word-gesture approach we implemented might not be the fastest way to input words. At least this can be said for beginners. But it seems, that it has a lower error rate by far compared to single letter input methods.

### 1.5.2 New Implementations

Because of the feedback and some observations, we decide to do one more implementation.
It is not a big change in the code, but can still be fairly useful. We have the problem, that for example, “thee” is prioritized before “the”, which does not make much sense, because “the” is used far more in plain language. Because the word list we are currently using is ordered by the frequency of the words, we decided to let this influence the best match and word suggestions. Now, if two or more words have the same highest score, the one that is highest in the word list will be taken, followed by the second highest and so on, until a word with a lower score comes next. This will prevent the problem with some words listed in Table 1.2.

# Bibliography


[1] Ahmed Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. pages 100 – 105, 2009.

[2] Costas Boletsis and Stian Kongsvik. Controller-based text-input techniques for virtual reality: An empirical comparison. *Int. J. Virtual Real.*, 19:2–15, 2019.

[3] Sibo Chen, Junce Wang, Santiago Guerra, Neha Mittal, and Soravis Prakkamakul. Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–6, 2019.

[4] I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, pages 754—-755, 2003.

[5] R. William Soukoreff and I. Scott MacKenzie. 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, page 113–120, 2003.

# A

# Appendix

University of Basel

Faculty of Science

## Declaration on Scientific Integrity

(including a Declaration on Plagiarism and Fraud)
Translation from German original

Title of Thesis:

Name Assesor: _____

Name Student: _____

Matriculation No.: _____

With my signature I declare that this submission is my own work and that I have fully acknowledged the assistance received in completing this work and that it contains no material that has not been formally acknowledged. I have mentioned all source materials used and have cited these in accordance with recognised scientific rules.

Place, Date: _____ Student: _____

---

Will this work be published?

☐ No

☐ Yes. With my signature I confirm that I agree to a publication of the work (print/digital) in the library, on the research database of the University of Basel and/or on the document server of the department. Likewise, I agree to the bibliographic reference in the catalog SLSP (Swiss Library Service Platform). (cross out as applicable)

Publication as of: _____

Place, Date: _____ Student: _____

Place, Date: _____ Assessor: _____

*Please enclose a completed and signed copy of this declaration in your Bachelor's or Master's thesis .*