# 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
0000-000-000

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 normal 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), with todays technology, it lacks 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 typing/ slide-to-type keyboards have been introduced in most major smartphone operating systems. Could this also be an efficient text input method for VR/AR?

# Table of Contents


**Acknowledgments** **ii**

**Abstract** **iii**

**1 Introduction** **1**

1.1 Motivation . . . . . 1

1.2 Goals . . . . . 1

**2 Related Work/ Background** **2**

2.1 vitrivr-VR and UnityVR . . . . . 2

2.2 Classical keyboard . . . . . 2

2.3 Word-Gesture Keyboard . . . . . 3

2.3.1 SHARK2 . . . . . 3

**3 Implementation** **4**

3.1 Functions of my Word-Gesture Keyboard . . . . . 4

3.1.1 Different functions is Subsections . . . . . 4

**4 Evaluation** **5**

4.1 MacKenzie Phrase Set . . . . . 5

4.2 Carry-out . . . . . 5

4.3 Results . . . . . 5

4.4 Discussion . . . . . 5

**5 Conclusion** **6**

5.1 Results Discussion . . . . . 6

5.2 Future Work . . . . . 6

**6 Body of the Thesis** **7**

6.1 Structure . . . . . 7

6.1.1 Sub-Section . . . . . 7

6.1.1.1 Sub-Sub-Section . . . . . 7

6.2 Equations . . . . . 7

6.3 Tables . . . . . 7

6.4 Figures . . . . . 8

6.5 Packages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

**Bibliography** **9**

**Appendix A Appendix** **10**

# 1 Introduction

## 1.1 Motivation

If we want to work with vitrivr-VR we have to put text into the query input text field. Right now, this is only possible with a normal keyboard, hence we have to tap on every single letter we want to write. Even though these ”texts” will only contain a single word or some few words, it still might be exhausting for our arms. The keyboard in vitrivr-VR has a bigger scale than a normal keyboard in reality. We have to move quite a distance with our arms and always move up and down to not accidentally hit a key. With a word-gesture keyboard, the text input could become more comfortable. We would not need to move our arms up and down, we could just move on a flat plane from one key to another. Such a keyboard could also be smaller, because the precision is not as important as it is for the normal keyboard. For example, if we would tap in the middle of two keys, we cannot really tell which one to take for the input. But with a word-gesture keyboard, where we work with distances and graphs (more on that later), it has not that much of an impact. Therefore, a smaller keyboard is possible, and we do not have to move our arms that much.
Other applications often do also only use a normal keyboard as text input method. The word-gesture keyboard developed in this thesis will be open-source and available for everybody. Therefore, developers of other VR applications may also be using our word-gesture keyboard if they are interested.

## 1.2 Goals

For this thesis, we have two main goals. The first one is to develop a word-gesture keyboard. This is a keyboard, that more or less might look like a normal one. But instead of tapping on the different keys, words are written with gestures. It has to work with vitrivr-VR and has to be available as open-source. It should also be available as a Unity package, so other developers can use it in their Unity projects as well. The second goal is to evaluate said keyboard. The evaluation will be conducted according to current research standards, and we use the MacKenzie phrase set.

# 2

# Related Work/ Background

In this chapter we introduce the environment the word-gesture keyboard is mainly developed for, some things about the normal, classical keyboard and word-gesture keyboard in general, with an excursion about SHARK2.

## 2.1 vitrivr-VR and UnityVR

Vitrivr[1] is an open source full stack content-based multimedia retrieval system. It supports image, audio and 3D collections and features a very broad set of query paradigms that are supported. Vitrivr was developed by the Database and Information Systems group[2] (dbis) of the university of Basel. For our thesis, we use the VR part of vitrivr, namely vitrivr-VR. This is being developed in Unity[3]. Unity is a tool for developers, where one can create projects in 2D, 3D and VR. To a certain degree, Unity is free to use, but one can provide assets that are not for free. In Unity, a developer can also provide packages. These can be imported and used by other developers in their Unity projects.

## 2.2 Classical keyboard

A classical keyboard is the most used keyboard type. On desktop and laptop computers we normally use such a keyboard. One thing that might be different in some countries is the layout. The layout does not change the functionality but can influence the appearance. This also applies to the most used keyboard for phones, tablets and other touchscreen-based devices. The only difference is that we do not press physical keys, but tap on the screen, where a certain key is. These keyboards work really well for text input with the previously mentioned devices. But when it comes to virtual reality (VR) or augmented reality (AR), this may not be the best possible text input method. Right now, it lacks of tactile feedback and accurate finger tracking. While this could be improved during the next years, yet it is not really there. Another reason is the size of such keyboards in VR. Due to the lack of

[1] https://dbis.dmi.unibas.ch/research/projects/vitrivr-project/
[2] https://dbis.dmi.unibas.ch
[3] https://unity.com

accurate finger tracking, we have to tap on the keys with our controllers. If the keys are too close together, it might cause a problem in recognizing which key was pressed. Therefore, there needs to be either bigger keys or bigger spaces between two adjacent keys. This results in a bigger keyboard, which results in more needed movement with the arms. If we have to move our arms a lot to input some text, this can quickly become exhausting.

## 2.3 Word-Gesture Keyboard

A word-gesture keyboard may look pretty much the same as a classical keyboard described in the last section, but works completely different. Independent of the details of the implementation, every word-gesture keyboard (also called slide-to-type keyboard) works with gestures. That means, instead of tapping on single keys, we have to draw one line or a shape on the keyboard. This will then be evaluated by an algorithm, that determines the closest word from a lexicon. Here closeness is determined by shape comparison between the user input and a word from the lexicon.

### 2.3.1 SHARK2

SHARK2 is a system introduced by Zhai and Kristensson (2003)

# 3 Implementation

This is the body of the thesis.

## 3.1 Functions of my Word-Gesture Keyboard

### 3.1.1 Different functions is Subsections

# 4 Evaluation

This is the body of the thesis.

## 4.1 MacKenzie Phrase Set

## 4.2 Carry-out

## 4.3 Results

## 4.4 Discussion

# 5 Conclusion

This is the body of the thesis.

## 5.1 Results Discussion

## 5.2 Future Work

# 6

# Body of the Thesis

This is the body of the thesis.

## 6.1 Structure

### 6.1.1 Sub-Section

#### 6.1.1.1 Sub-Sub-Section

**Paragraph**

**Even Sub-Paragraph** This is the body text. Make sure that when you reference anything you use labels and references. When you refer to anything, you normally capitalise the type of object you reference to, e.g. Section 6.1 instead of section 6.1. You may also just use the `cref` command and it will generate the label, e.g., for Section 6.1, we did not specify the word "Section".
Hint: Try to structure your labels as it is done with `sec:my-label` and `fig:machine`, etc.

## 6.2 Equations

A Turing Machine is a 7-Tuple:

$$M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle \tag{6.1}$$

A Turing Machine is a 7-Tuple even if defined in the text, as in $M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$.

## 6.3 Tables

Some tables can also be used as shown in Table 6.1[4]. Remember that tables might be positioned elsewhere in the document. You can force positioning by putting a `ht!` in the definition.

[4] Table captions are normally above the table.

Table 6.1: Frequency of Paper Citations. By the way: Make sure to put the label always after the caption, otherwise LaTeX might reference wrongly!

| Title | $f$ | Comments |
|---|---|---|
| The chemical basis of morphogenesis | 7327 | |
| On computable numbers, with an application to the ... | 6347 | Turing Machine |
| Computing machinery and intelligence | 6130 | |

## 6.4 Figures

Figures are nice to show concepts visually. For organising well your thesis, put all figures in the Figures folder. Figure 6.1 shows how to insert an image into your document. Figure 6.2 references a figure with multiple sub-figures, whereas the sub-figures are referenced by Fig. 6.2(a), etc.

Missing: Description figure.

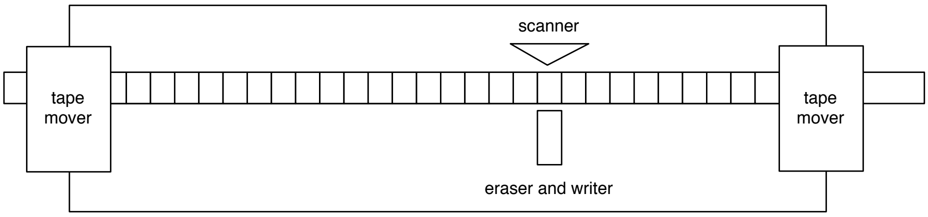
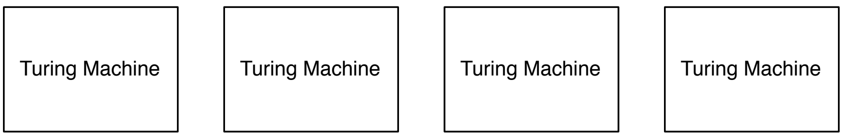



Figure 6.1: A Turing machine.




(a) Turing Machine 1 (b) Turing Machine 2 (c) Turing Machine 3 (d) Turing Machine 4

Figure 6.2: Plots of four Turing machines

## 6.5 Packages

These packages might be helpful for writing your thesis:

**`caption`** to adjust the look of your captions

**`glossaries`** for creating glossaries (also list of symbols)

**`makeidx`** for indexes and the back of your document

**`algorithm, algorithmicx, algpseudocode`** for adding algorithms to your document

# Bibliography

# 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 .*