RWTH Aachen University

Master Thesis

Text Input System for a Sensitive Data Glove with Haptic Feedback

by

Adam Brunnmeier

RWTH Aachen University

Master Thesis

Text Input System for a Sensitive Data Glove with Haptic Feedback

for the degree of M.Sc. in Computer Science

by

Adam Brunnmeier
Student Id.: 331 624

Prof. Dr. Torsten W. Kuhlen
Visual Computing Institute

Prof. Dr. Leif Kobbelt
Visual Computing Institute

Supervisor: Martin Bellgardt, M.Sc.

Date of issue: August 27, 2019

Statement

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig im Rahmen der an der RWTH Aachen üblichen Betreuung angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I guarantee herewith that this thesis has been done independently, with support of the Virtual Reality Group at the RWTH Aachen University, and that no other than the referenced sources were used.

Aachen, August 27, 2019 ....

Contents

Chapter 1

Introduction

HMDs hit the consumer market with an image resolution and refresh rate that matches current business- and consumerclass computer and smartphone displays. HMDs require an alternative to the keyboard for text input. For hand-held devices, the touch keyboard is the prevalent system. Though, the touch keyboard's performance is worse than the keyboard's one. Furthermore the touch-keyboard occupies a large portion of the display area and splits attention between the visual content and the visual typing feedback. The hardware-keyboard actually has also drawbacks in its enforced body-position and muscle-strain. This thesis evaluates a possibility to build a better text input for HMD-use. One very performant and intuitive way sure is dictation. But for the silent use case we optimally need a high-performant (measured in WPM), tactile-based (to release visual and aural sense), little-strain (for prolonged use), mobile device.

This thesis examines a data glove for this use case. Potentially a data glove has a very rich input range and excellent tactile feedback possibilities at a sensorially wealthy zone of the human body. With these preconditions and a little bit training it can provide high-dimensional input and output. It still didn't get the status of a mainstream interface yet. This work evaluates glove typing with the concrete data glove from Aachen's Cynteract GmbH with $< 0.01°$ precise orientation sensors and fast tactile feedback ($< 20ms$ without prediction) by letting people use it to type text with a rudimentary input system. The glove's firmware was slightly adapted for this use case.

When developing an alternative text input system, significant advantages over existing options must exist to motivate the effort of relearning a new typing system. Digital text input is trained from early years and several months to years are needed to gain expertise in one type of input system. Furthermore text is written on computer or mobile devices very often a day, proving a switch of input system heavily exhausting. On the other side, the extensive need for everyday text input which won't decrease in the foreseeable future is a motivator to explore alternative, more performant systems, even if it comes with an initially high effort to get used to. A known example where users put a lot of effort into relearning a text input system for performance reasons are the dvorak and neo keyboard layout. Another less intrusive example of using a new input system is Cynteract's glove usage during hand muscle rehabilitation. In this case, using the glove as input system for a computer game is just a bonus to the practical rehabilitation exercises, which require the hand movements anyways. When looking for use cases where the glove input can be used as an added bonus, many possibilities are found, e.g. enabling the user to choose a name for some fictional character in a game. This is a non-exhaustive procedure even with a new input system. Another example would be using the same motoric movements for user interface navigation that will also be used for text input later on. Due to the very high learning costs, a new typing system should be introduced with care and if the advantages aren't sustainable, promoting it is not useful.

A data glove is a rather simplistic input device without much processing involved in its basic form. The algorithmic task of a glove is to derive the bending angles of all fingers' joints. The two most characteristic parts of a glove are its sensors and its feedback devices. The sensors are used to measure angles. They are in most cases resistive flex sensors or inertial measurement units. More specialized algorithms can be used to fuse multiple sensors' data to improve inter-sensor accuracy. Apart from the sensors there are the builtin haptic feedback devices. From the perspective of power consumption, most power is drained by haptic feedback and by wireless communication due to the rather large amount of glove input data consisting of up to $5 * 3 + 1 = 16$ high precision rotation vectors at a high update frequency up to 1kHz.

A glove has potentially an input dimension of wrist (3), thumb (2+1), index finger (2+1), middle finger (2+1), ring finger (2+1) and pinky finger (2). That is a total of 17 dimensions. With this background, a rich diversity of gestures can be exploited for user input. Furthermore that leaves room to substitute movements of one part of the hand with another part of the hand. So there is a buffer for custom (re-)configurations that suit the needs of people with situational constraints or disabilities.

The input system in this thesis was developed iteratively, based on the Cynteract glove with two IMUs and a vibrational feedback device per finger. During development a new hardware prototype was assembled to meet latency requirements. Several small tests were performed periodically where people tried the system and their feedback went into making the system more usable. The optimization criteria for development are in decreasing priority performance, usability, strainlessness, learnability, intuitiveness. Performance is measured in words per minute against the keyboard. The test presented in the section "Evaluation" is the last feedback collected during this thesis. The test focused on measuring performance and intuitiveness. To summarize, the current system reached an input velocity of 40% of equivalent untrained keyboard velocity (=4% of trained keyboard velocity), an input accuracy about 30%, about half of the participants didn't need any explanation while using the glove, about half of the participants got along with blind typing and all complained about strain in the pinky finger. As the thesis goals were not met by the developed system yet, there are plans to continue the work until either the performance of the keyboard is reached or there is strong evidence that this will be harder than anticipated. There are still many iterations planned on the roadmap to improve in the optimization criteria.

The glove that was used for the implementation of the text input system is the device developed and built by Cynteract GmbH in Aachen, Germany. The glove's development started several years ago, was promoted by Germany's youth programme "Jugend forscht" and received several startup supports. It is not sold on the consumer market yet, as certifications are currently going on, which are the last step before the first limited distribution. Product development started with VR games as target and later set the focus on the application in clinical rehabilitation. As certification and quality assurance for the clinical application takes a lot of time, the game market again became an earlier target.

A suitable input system for a glove requires implementation of the three components hardware, firmware and a frontend software-library. In this case the firmware is an arduino-program running on an esp32. The frontend software is a C# library supplemented with samples implemented with the Unity3d game engine. The text input system induced latency and haptic feedback precision requirements that weren't met before so that all three components were modified to meet these requirements. A sufficient input precision was already required and met before by the application in rehabilitation.

People first using the glove are mostly curious. In the tests, they self-motivatedly discover the glove's interface behaviour. They are positively impressed when showing them an even imperfectly aligned simulated hand model. When the glove is used as input device for a computer game the interaction methods are learned with fun.

A rather significant drawback during the development of the text input system was the haptic feedback latency. The hardware needed modification to guarantee a sufficiently low vibrational feedback latency. Luckily the team from Cynteract already planned building a new prototype so that they incorporated faster haptic feedback devices in a new prototype. But the system so far was iterated with slow haptic feedback, the last evaluation in this thesis uses the new latency-improved prototype the first time. The latency was only affecting haptic feedback, visual feedback was not affected and also mainly used during intermediate evaluations. Furthermore the prototype used for development didn't implement recognition of the pinky finger so that evaluation of simultaneous usage of all four fingers is firstly done in the last evaluation with the new prototype.

Chapter 2

Related Work

The globally prevalent text-input devices in 2019 are the keyboard and the touch screen. While the keyboard borrowed its design and mechanics from the typewriter, the touch screen is mostly used with a virtual one-touch-one-letter keyboard with a rather similar design as its hardware counterpart. There are variations like one-handed keyboards[1] or swiping touch input. An alternative device is a data glove, which is also the type of hardware used in this thesis. There are commercial appliances of data gloves reaching from the NES PowerGlove from 1987 with flex units to gloves with IMUs, vibrational feedback and force feedback [2]. The 5DT glove also provides mouse emulation[3].

The next sections in this chapter will first deal with academic works that evaluated the construction of a data glove. After that, some work is presented that enables sign language input with a glove. The last section in this chapter deals with other works that implement a sign language or, more generally, a hand gesture recognition system.

There are of course also other text input alternatives than keyboards, touchscreen keyboards and data gloves. A significant one in terms of performance and usability is text dictation. It isn't considered in this thesis because of the noise it generates and thus omitting some use cases. A silent dictation system would change this consideration of course.

Another academic work dealt with handwriting in the air and used an alphabet of 27 symbols with 80% accuracy ([AS12]). They built a glove (Figure 2.1) that measures hand motion with an accelerometer and a gyroscope which are attached to the back

[1] e.g. Frogpad, https://en.wikipedia.org/wiki/FrogPad
[2] e.g. CyberGrasp, http://www.cyberglovesystems.com/
[3] 5DT Data Glove 5 User's Manual

Figure 2.1: Glove used by [AS12]

of the hand. The sensor values are sent wirelessly to the processing computer. The interface enables a user to input text into a computer by writing in the air like on an imaginary blackboard. The system continuously recognizes arbitrary sentences based on a predefined vocabulary in real-time. The recognizer uses Hidden Markov Models (HMM) together with a statistical language model. They achieve a word error rate of 11% for a 8K vocabulary based on an experiment with nine users.

The commercial Leap Motion device is analyzing hand movements with camera images. Some people developed a virtual keyboard for it[4]. There is also a technically equivalent academic version [LW03]. They develop a 3d augmented reality keyboard (Figure 2.2) which enables a user to type text or control CG objects. Their used camera is stereoscopic and attached to an HMD. The system consists of three modules: 3d vision-based tracking, natural interaction with the fingers, and audiovisual feedback on the 3d video see-through HMD.

2.1 Glove Device

This section deals with glove devices built and evaluated in academic context.

[CYKW18] designed and built a hand tracking glove (Figure 2.3) that is able to track

[4] https://developer-archive.leapmotion.com/gallery/vr-keyboard

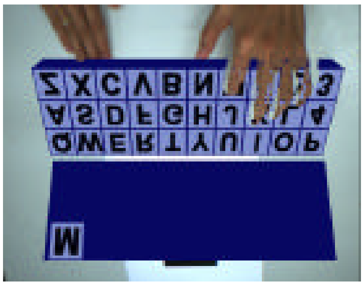


Figure 2.2: Augmented keyboard used by [LW03]

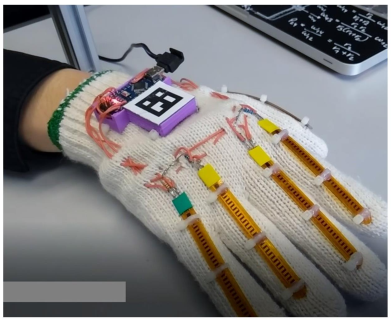


Figure 2.3: Glove used by [CYKW18]

the pose of the hand and the motion of the five fingers. They synthesized the input from a composition of 5 flex-sensors, a 3-axis IMU and a camera. They attached a ArUco marker to the back of the glove to obtain the pose information of the hand from the camera. They applied a kalman filter to stabilize the input. To increase the sampling rate up to 100Hz they added an IMU. The system fuses the values from the separate sensors. While the ArUco marker is occluded temporarily, the pose of the glove can still be obtained. They use the flex sensors to track the finger motion. They developed a virtual hand model in the computer that moves simultaneously with the physical human hand.

[SAW+14] presents an algorithm to convert flex-sensor values to angles. They built a glove (Figure 2.4) that measures finger movement. Their work focuses on the algorithm used to correlate the voltage outputted from the data glove with the finger bending angle. They use a polynomial regression analysis to transform the voltage output from the flex sensors into an angle. They claim a voltage to angle conversion accuracy of 100%.

[SNSK18] built a glove with 5 flex-sensors, a 3-axis accelerometer, an Atmega μC and a wifi-module. They used it to recognize Kannada sign language. The system classifies the hand sign and transforms it into final text and speech. The system is portable and connects to any android platform. It allows someone with no knowledge of the Kannada sign language to be able to understand and interact.

Figure 2.4: Glove used by [SAW+14]

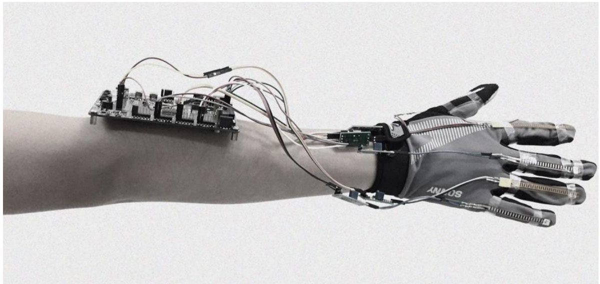


Figure 2.5: Glove used by [XZL+18]

[XZL+18] used an STM32 including a wifi module, a 3-axis gyroscope and flex sensors in their glove (Figure 2.5). Their goal was to enhance learning interest of deaf-mute children as beginners in sign language and promote educational balance. They designed their application to be enlightening for children learning sign language. The application consists of the data acquisition and conversion, the wireless communication and the interactive cartoon frontend. They claim that their system has high accuracy in gesture recognition and a reliable response time. Deaf-mute children can use the cartoonish application to get sign language examination questions and to answer them by wearing data gloves. The system will display a star rating based on whether the answer is correct and on whether the cue was clicked. Their work attracted interest from the Qingdao school for deaf children.

[TL03] added a force sensor to their custom built glove (Figure 2.6). They used a rubber-coated cotton glove. The sensors are firmly attached to the rubber-coated glove using cyanoacrilic glue. The force sensor is made of a steel plate substrate. The flex sensors are attached to it. The plate is attached on the thumb. The force sensor responds linearly with a resolution of 0.38 N and a sensitivity of 0.05 V/N.

[BSF14] used the commercial CyberGrasp (Figure 2.7) to test muscle reinforcement for elderly people. They are motivated by the significant increase of over 60-year-old population, which will reach nearly 2 billion in 2050 according to WHO. They used the gloves to simulate everyday activities in a virtual environment. During these simulations they measured the levels of manual strength of elderly people. The activities were chosen according to the group's usual everyday life. In a second part they evaluated hand

Figure 2.6: Glove used by [TL03]

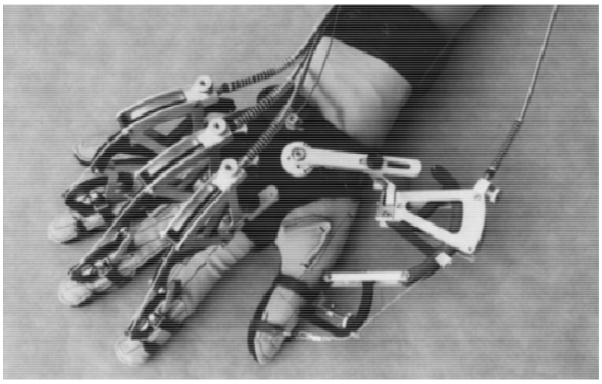


Figure 2.7: CyberGrasp used by [BSF14]

reinforcement with cables connected to the fingertips for hand-impaired use cases.

2.2 Sign Language

Many of the documents found on google scholar which include a data glove, deal with the appliance of a sign language for deaf. This paragraph tells about the more progressed part of those works and its used hardware and software as well as the performance reached.

The greatest alphabet size was achieved by [LO98] with 250 distinct expressions and a detection accuracy of .7. It used two gloves (model "DataGlove") and a separate Polhemus 3D tracker as hardware and an HMM as algorithm. Their recognition happens continuously in real-time. They identify the endpoint-detection in a stream of gesture input as most critical problem. The system's statistical analysis is done according to the

Figure 2.8: Glove used by [KTG+04]

4 parameters posture, position, orientation, and motion of a gesture. They implemented a prototype system with a lexicon of 250 vocabularies in Taiwanese sign language. The system uses HMMs for 51 fundamental postures, 6 orientations, and 8 motion primitives. In a user-dependent way, a sentence of gestures based on these vocabularies can be continuously recognized in real-time and the average recognition rate is 80.4%.

[CGM00] used two CyberGlove's and a Polhemus 3D tracker as hardware and achieved $n = 65$ with accuracy $p = .86$, using a decisiontree-algorithm. They derive some rules from the decision tree using training data of 65 different hand gestures. The sensor data is being normalized for all sensors to accommodate the variations of each sensor.

[KTG+04] used a self-built pair of low-cost gloves (Figure 2.8), they name it StrinGlove. The glove uses 24 Inductcoders and 9 contact sensors. It encodes hand postures into posture codes on its own processor unit. They evaluated their prototype with several sign experts. Later ([TKO10]) they worked on finding a minimal number of sensors for a given set of discrete hand postures. They applied their method to Japanese sign language and concluded that a glove with just eight sensors can obtain all hand postures given in Japanese sign language.

[MT07] used a data glove with a recurrent neural network to recognize Japanese sign language. First they implemented a posture recognition system which could recognize a finger alphabet of 42 symbols. Then they added continuous gesture recognition.

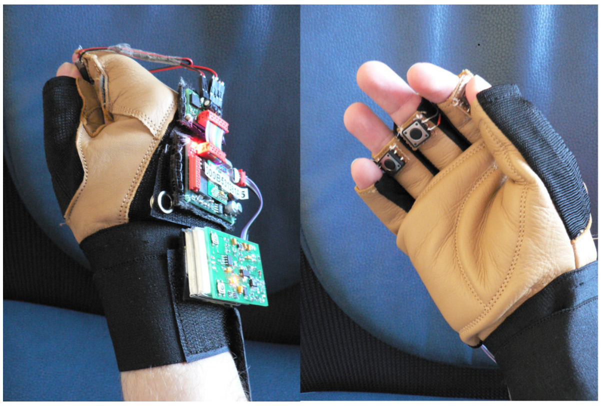


Figure 2.9: Glove used by [WJ07]

2.3 Text Input

This section deals with the (gesture) input systems that didn't reach much performance. They all have an input alphabet with at most 34 elements.

[WJ07] used an alphabet of 34 letters for text input with a self-built glove (Figure 2.9) and reached a WPM between 2 and 3 after 100 minutes training, using input composition sequences. They design and initially evaluate two methods for short text input using hand gestures. Their wireless glove is able to recognize 4 basic gestures and is used together with the chording principle. They evaluate two different concepts to map gestures to characters. Their preliminary experiment results show that simple free hand gestures in combination with different key maps are easier to learn and allow faster typing than distinct gestures assigned for each character.

[CD12] used a learning vector quantization algorithm with the DG5 VHand glove (Figure 2.10). Their gesture recognition algorithm works in real-time. One part of it is a data glove which performs the feature extraction. The second part is the classifier which applies learning vector quantization. The recognizer was tested on a dataset of 3900 hand gestures and reached a recognition rate of $> 99\%$ for a 13-element large alphabet.

[CFH03, KPBB02] both used a camera based system with a HMM algorithm. [CFH03]'s system recognizes continuous hand gestures in front of an unchanging background. The system consists of four modules: real time hand tracking and extraction, feature extraction, HMM training, and gesture recognition. First, they apply a real-time hand tracking and extraction algorithm to trace the moving hand and extract the hand region, then they use the Fourier descriptor to characterize spatial features and the motion analysis to characterize the temporal features. They combine the spatial and temporal features of the input image sequence as their feature vector. After extracting the feature vectors, they apply HMMs to recognize the input gesture. The gesture to be recognized is sep-

Figure 2.10: DG5 VHand used by [CD12]

arately scored against different HMMs and the model with the highest score indicates the corresponding gesture. In their experiments, they tested the system to recognize 20 different gestures and got a recognition rate of above 90%. [KPBB02] tries to recognize continuous Korean sign language. While recognizing gesture words in a sign language, one problem is to segment a continuous sign into individual sign words. To solve this problem, they disassemble the sign language into 18 hand motion classes according to their patterns and represent the sign words as some combination of hand motions. Observing the speed and the change of speed of hand motion and using fuzzy partitioning and state automata, they filter unintended gesture motions such as preparatory motion and meaningless movement between sign words. They recognize 18 hand motion classes with a HMM. With these methods they succeed to recognize 15 sentences with 94% recognition ratio in their evaluation.

[EC14] used a regular camera combined with three differential pyroelectric infrared sensors (Figure 2.11) for hand gesture recognition. Movements within the viewing range of the infrared sensors are first detected by the sensors. When they detect a change, they trigger the image-based hand gesture analysis. If the movement is due to a hand, one-dimensional continuous-time signals extracted from the infrared sensors are used to classify/recognize the hand movements in real-time. Classification of different hand gestures by using the infrared sensors is carried out by a winner-takes-all hash based recognition method they developed. Jaccard distance is used to compare the hash codes extracted from the one-dimensional differential infrared sensor data. During evaluation they recognized 5 different gestures. The additional use of infrared sensors complementing the video camera increases accuracy and saves power.

[FJSD13] used a camera to recognize two handed sign language. After applying the preprocessing steps cropping, rotation, colour filtering and resizing, the mapping is done using a correlation coefficient process and a nail detection process. The accuracy of the system was tested in real-time in a setup using MATLAB, a 1MP webcam and a greenscreen. The system recognizes signs from a 10-element alphabet with an accuracy

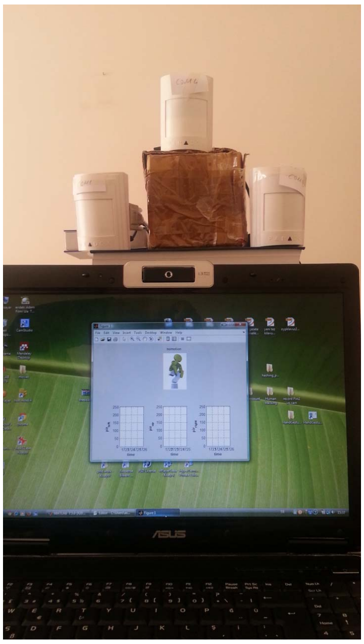


Figure 2.11: Videocamera and differential pyroelectric infrared camera used by [EC14]

between 40% and 80%.

[FR94] used a camera based system with an algorithm using histograms of local orientation. They use the orientation histogram as a feature vector for gesture classification and interpolation. It is simple, fast to compute, and has some robustness to scene illumination changes. They implemented a real-time version, which can distinguish 10 different hand gestures. They developed a sample software where a user can control an animated crane with hand gestures or play a game. The gesture vocabulary must be constructed omitting gestures with similar orientation histograms.

[GSK12, LCoS11] both used a camera based system using an FPGA. [GSK12]'s real-time system recognizes 10 different hand gestures. The gestures are classified on basis of shape-based features. Four different shape based features are used for better accuracy. An illumination compensation technique is employed for robust recognition under varying background lighting conditions. Skin color segmentation is used to minimize the chances of false detection. They modeled their system using Verilog HDL and a Xilinx FPGA board. Their test reached a recognition rate of 94.40%. [LCoS11] uses an Altera FPGA board to build a real-time gesture recognizer and hand tracker. The hand tracking system includes the image preprocessing and subsequent feature extraction consisting of bounding box and center-of-mass computation. The extracted features are used for gesture sign classification. The main modules in the image processing step exploit the parallel architecture of the FPGA to achieve real time processing.

[LM13] built a wrist-worn 3-axis accelerometer sensor for gesture recognition. Their

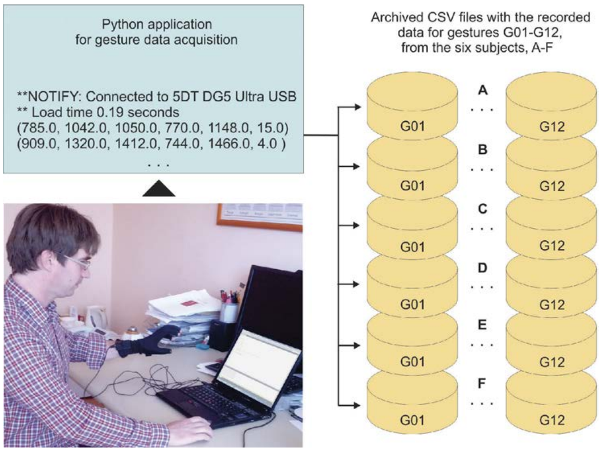


Figure 2.12: 5DT Data Glove 5 Ultra used by [LP14]

system can recognize eight hand gestures based on the accelerometer. The gestures are classified by comparing the acceleration values with stored templates. They developed an application that displays commands on a LCD and plays sound depending on the recognized gesture.

[LP14] used the 5DT Data Glove (Figure 2.12) with 5 one-dimensional sensors and an algorithm using clustering and a probabilistic neural network. The clustering is done to compress the training data and accelerate neural network training and recognition. The sensor data is clustered separately with three separate algorithms: k-means, x-means and an expectation maximizer. In the training process each gesture class is globally and permanently assigned the least error-causing of those 3 static clustering algorithms which is then also exclusively used during recognition for this gesture class. In their evaluation with 12 distinct gestures they reached 97% accuracy for people who contributed to the training set and 86% accuracy for a subject who did not contribute to the training set.

[PFN$^+$08] used a wrist-worn camera and finger mounted dice as markers (Figure 2.13). The camera tracks the visual markers at the finger tips and software on the PC computes the position of each finger tip and its joints in real-time. In their evaluation they measured a mean position-detection error of 0.6mm.

[SAP16] used a camera with a neural network algorithm. The image from the camera is denoised, color filtered and resized, then it is segmented with help of k-nearest neighbors algorithm. From the resulting segmented image the features extrema points, area, centroid, diameter and perimeter are extracted. Recognition and previous training is then done with a pulse coupled neural network.

[Smi13] developed a system to recognize body gestures with a camera and a neural network. They use a color tracking algorithm together with coloured clothes. They segment the image and calculate the center-of-mass for the left hand, the right hand

Figure 2.13: Glove used by [PFN+08]

and a marker on the chest. These points are sent through a feed-forward neural network and trained using a backpropagation algorithm. Training is done over an ensemble using different randomized initial weights to reduce local minima error. They evaluated their system on 4 distinct gestures and reached 98% recognition rate.

[TSMM17] used swiping gestures on a smartphone and processed them with a HMM algorithm. They show that noise generated by human gestures and captured by the sensors of those devices degrade training and classification accuracy for gesture recognition in state-of-the-art deterministic HMMs. They use another statistical quantization process that mitigates these problems. During training they produce gesture-specific codebooks, HMMs, and error models for gesture sequences. During classification they exploit the error model to explore multiple feasible HMM state sequences. They implement classification in *Uncertain(T)*, a probabilistic programming system that encapsulates HMMs and error models and then automates sampling and inference in the runtime. *Uncertain(T)* lets them choose an application-specific trade-off between recall and precision at gesture recognition time, rather than at training time. They evaluate an average recognition rate of 71% for 20 swiping gestures.

[WP14] used a wrist-worn camera and microphone (Figure 2.14) to detect small-movement hand gestures. The images were classified with a bag-of-words support vector machine model. They used the microphone to detect vibrations made by the thumb and finger during a pinch gesture, and utilize the vibration for accurate recognition.

[WS99] used the CyberGlove with a neural network algorithm. The glove provides 18 measurement values for the angles of the finger joints. The authors compare the performance of three variations of back-propagation neural network and a radial basis function neural network on a set of 20 gestures. Their recognition rate during evaluation reaches 100% in several configurations.

[YYQ+18] used a custom glove (Figure 2.15) and developed an algorithm based on a vector-template-distance and a neural network, optimizing for low processing-power. The glove uses flex sensors and a STM32 processing unit. Their algorithm combines a

Figure 2.14: Wrist-device used by [WP14]

Figure 2.15: Glove used by [YYQ+18]

template matching step with a back-propagating neural network. First the data obtained from the flex sensors is matched against specified templates. The outcome is fed to the neural network to recognize the gesture. The template matching shrinks the amount of data put into the neural network and thus decreases the needed processing power. In their evaluation of recognizing 9 gestures in real-time they reached a recognition rate of 100%.

CHAPTER 3

USABILITY CONSIDERATION

This chapter deals with requirements and limitations of using a glove-driven text input, with usability, performance and ergonomic considerations and to which extent they are addressed with the given glove and input system.

3.1 Separation of Keyboard and Mouse

People use a mouse with a desktop computer, a trackpad or fingerstick with a laptop and the planar touchscreen with a smartphone as 2d pointing device to navigate 2d GUIs. For HMDs 2d GUIs currently are a relevant part, too. If a data glove is used as typing device, it must also provide 2d pointing capabilities to prevent the need to switch the device. While in a VR environment the stretched index finger may be sufficient, for all other usecases there must be a mouse-equivalent functionality. That is the reason the thumb was spared in this thesis, to be able to potentially use it as pointing device. Though implementing a mode switching feature would render this consideration useless, as mouse and keyboard are almost never used simultaneously. Implementing a mouse functionality was already done by 5DT Data Glove 5, it is described in its user manual.

3.2 Hardware Compatibility

A typing glove can be used for desktop computers, laptops and smartphones. The glove used in this thesis currently interacts with the client computer only via USB serial device and bluetooth serial device protocols, thus for typing it needs external software. But there is no technical limitation that prevents implementing the conventional bluetooth/USB HID (Human Interface Device) compliant 2d pointing device and keyboard protocol, enabling plug and play on any device. For usage of features not covered by standard mouse and keyboard, still platform-dependent software must be maintained.

3.3 Symbol Alphabet

For typing tasks, the given input system must be able to insert at least the alphanumerical and punctuation symbols in a given language. Phonetically based alphabets typically have around 30 letters, the Japanese syllable alphabet has around 200 elements and the Kanji word alphabet has around 2000 active elements. Including capitals ($*2$), one or two modifier layers ($*3$) and optionally a multi-language layout, about 300 input possibilities for symbols must be provided. The ~50 remaining printable ASCII characters add to that. To find out the number of symbols which should be possible to input for a general use system, optimally every possible use case should be considered. For navigation purpose, the directional symbols next/previous/first/last are added, with modifiers to navigate characters/tabs/history/windows/workspaces in 2 dimensions and modifiers to move, mark and shift. That adds 120 symbols. Furthermore 30 control-symbols like escape/delete/print/brightness are accounted for. Iconic symbols like emojis, block drawings or arrows are estimated with ~400 symbols but are deferred to compositional input as combination of other symbols. In Linux this is known as "Compose Key". Everything adds up to around 500-600 distinct input symbols. By distributing the input dimensionality evenly among every of the four used fingers, each finger must be able to input at least 5 distinct, discrete values ($5^4 = 625$). If using state-based input, the number of unique inputs can be significantly reduced. If different languages and navigation are split into different operation modes, 230 distinct input gestures are sufficient.

3.4 Mobile Use

Not so relevant in combination with computer monitors but more so with a laptop or even a smartphone/HMD in transit, the system requires wireless operation. The glove used in this work supports bluetooth. To cover even more usecases, a wireless USB receiver should be added. For the transit usecase, it is also beneficial to be able to do input while carrying something in the hand or while clinging to something. While the glove works single-handedly, a complementing pressure-based input should be evaluated, too. That could potentially enable carrying or clinging with the hand used for input.

Another aspect when considering mobile usage is energy consumption. If the device is operated wirelessly it draws current from its battery, if it is operated via USB cable it will draw current from its host. Therefore as little power consumption as possible is preferred. Almost all of the power consumption is caused by wireless communication and the vibrational feedback devices. To minimize those, BLE and relatively low amplitude LRAs are evaluated.

3.5 Strain

Usage both at work and privately requires application with almost no strain for prolonged use of several hours. When you use a monitor or a HMD, a long-wired as well as a completely wireless data glove is not requiring a specific body position except that your fingers must be freely movable. Input gestures shall require minimal movement and minimal applied force to counteract fatigue and increase input speed. The condition to have freely movable fingers can lead to tension in the wrist while holding the fingers in the air. Furthermore, the current implementation applies a static base position which can lead to constant tension in the fingers while holding them in that position. The condition to have freely movable fingers can be possibly avoided by complementing the system with a pressure-based input using the force-sensors at the fingertips but it wasn't evaluated in this work. The finger-tension problem can be tried to be solved by implementing an adaptive base position or a differential input system. This work implemented a calibration process activated by a button press and requiring the fingers to swing in the desired operative angle range several seconds to complete. It works but pressing a button is not convenient, it should be replaced with some gesture like touching a finger with the thumb.

Another annoying disadvantage of many data gloves and the specific glove used in this work is that the fabric glove interferes with skin cooling, diminishing comfort of use and necessitating hygienic considerations. This problem can be reduced with a fabric that supports skin breathing and by designing cut-out regions to reduce the surface of fabric contacting the skin.

CHAPTER 4

USED INPUT SYSTEM

This chapter mentions the used hardware, lists the requirements for high-performant typing and elaborates on the system that turns finger movements into characters, with and without visual aid.

4.1 Hardware

The hardware (Figure 4.1) includes the fabric glove with covered electronics, a USB cable and the client PC. The glove is a pre-release version designed and built by Aachen's startup Cynteract GmbH, founded by students. The glove's electronics (Figure 4.2) has an ESP32 microprocessor with integrated wifi and bluetooth features, two IMUs (inertial measurement unit) per finger and one on the hand's back, one force sensor on each fingertip, one tactile feedback device on each fingertip and one on the hand's back.

4.2 Reaching High WPM

To reach high WPM-performance there are some requirements to the input system. For comparison, here are some citations from the ANSI BSR/HFES 100 standard regarding keyboards:

- buttons should Be activated with thumb/finger flexion, not extension

Figure 4.1: The last but two and last but one prototype from Cynteract

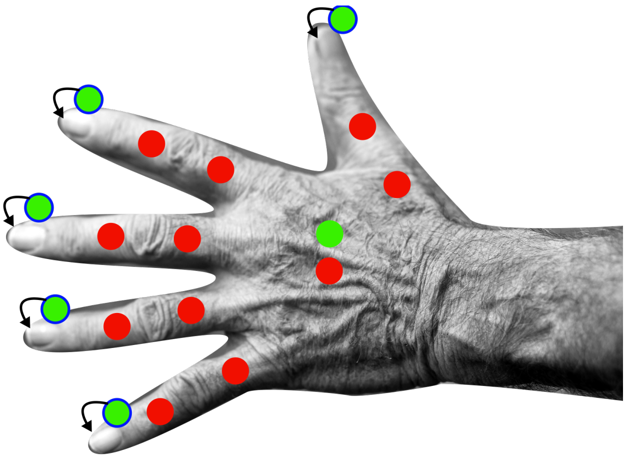


Figure 4.2: Glove scheme: 9-axis inertial measurement unit red, tactile linear resonant actuator feedback green, force sensor blue contour.

- maximum force needed to press and activate such buttons shall Be between 0.25 and 1.5 N
- buttons should Have a displacement between 1.0 and 6.0 mm
- buttons should Provide feedback to users upon activation
- the point of key activation should Be marked by a distinguishable breakaway force
- Elimination of the breakaway force or the subsequent cushioning force can result in slower keying activity, higher error rates, and increased operator fatigue

Adequately the data glove's input system shall use inherently strong muscles, have rather small movement vectors and provide tactile feedback on symbol selection as well as symbol input. As no physical buttons are used, care must be taken that the tactile feedback latency is as small as possible. A common used boundary in VR research is the requirement of $< 1ms$ latency. This latency is not reached in this work's input system for two reasons. First, the sequential reading of the 12 IMUs has a duration of 20ms, limited by the 400kbit/s I^2C fast mode speed. Second, the start-up time of the vibrational feedback to a perceivable value lies between 5ms and 30ms.

4.3 Software

The software was partly implemented on the embedded ESP32-chip and partly as Unity3d script using the glove's software library. Unity3d is a popular game engine and was chosen because the software library for the glove was written in it. When using the PC for $< 100ms$ latency-sensitive applications, the scheduling quantum will get relevant, as it most often is greater than $10ms$. A simple program (implemented twice, in python and in C#) added 45ms latency when using non-blocking communication with the glove on Windows 10. Using polling in two blocking extra-threads reduced the latency to $< 1ms$.

4.4 Mapping a 3d Hand-Model

For the typing usecase, a very precise shape approximation of the hand is not required, it is sufficient if the single sensor-resolution is high enough. The IMUs deliver an absolute orientation, aligned with the earth's magnetic field, and acceleration. The acceleration is not used in the glove yet. The absolute orientation is mapped to a hand model consisting of the five fingers attached to the palm. Each finger is itself divided into three parts, the root section, the middle section and the end section with the fingernail. In almost all cases, the end section's rotation is directly derivable from the middle section's rotation. That is why the IMUs in the end sections were spared in the data glove. Based on the hand model, the IMUs' raw orientation data is transformed to bending angles of each finger, which in turn can be used to animate a 3d-model or for input gestures.

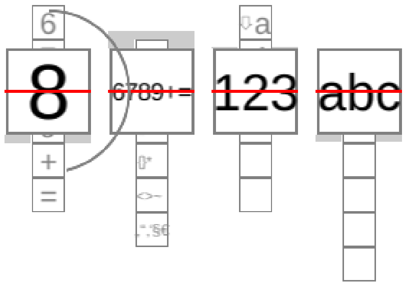
The raw orientation data is a set of quaternions and its transformation to bending angles requires a calibration step, as the sensors have a small locational and rotational displacement on the skin's surface. The calibration procedure is not yet sufficiently solved, there are easily recognizable discrepancies between the physical hand and the virtual representation.

4.5 Symbol Input Mechanics

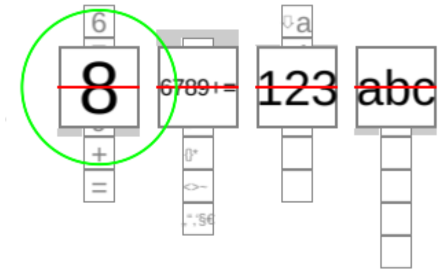
This section describes the practical process of inputting text. This is how it works: You bend and extend your index, middle, ring and pinky finger into a specific position to select a symbol. When resting in that position for approximately 1s, the selected symbol is entered. The thumb is not used for input. Every finger can be bent into 6 distinct, equal-size, consecutive angle-intervals. If the finger is outside any of those 6 zones, it isn't considered for input. In tests with first-use participants a maximum resolution of up to $2° - 4°$ per zone and a maximum frequency of up to $500ms$ could be reliably controlled by the user's motoric system. Any higher resolution or frequency causes input-inaccuracies without any training. With 6 distinct zones per finger there is a technical input range of $(6 + 1)^4 = 2401$ different symbols.

The tactile feedback consists of a short "click" vibration on zone-boundary traversal per finger, a permanent vibration pattern inside a zone per finger and a "click" vibration on symbol input on the back of the hand. There are three variations of zone-boundary traversal "click" vibrations, assigned to the cases of switching from a higher zone into a lower zone, switching from a lower zone into a higher zone and switching from one of the 6 active zones into an inactive angle. There are also three different vibration patterns inside the zone, one for zone 1+4, one for zone 2+5 and one for zone 3+6. The zone traversal and zone residence feedback enable spatial navigation, the backhand input feedback enables temporal navigation.

The visual feedback consists of four vertical bars (Figure 4.3), each representing one finger. Bending a finger will move its bar down, extending the finger will move it up. The bar is divided into 6 equally-sized squares, each square representing one zone. The zone at a visually outlined vertical level is the currently selected zone. A loading circle in the currently selected zone indicates the rest time, when it reaches ~1s, the selected symbol is entered and the loading circle flashes in another color. Inside each zone-square there is a visual representation of the corresponding input, e.g. a letter or a digit. With these representations the user can browse the whole symbol map.



(a) Each finger controls a column.



(b) After resting a specific time (~1s) inside a zone-configuration, the symbol is typed.

Figure 4.3: Visual feedback.

4.6 Intermediate Results

In the beginning of implementation the glove had a high input and feedback latency due to a configured low serial baud rate, low I^2C baud rate, process scheduling issues in the client PC's software, slowly responding vibrational feedback motors, uncompressed communication and a program flow in the glove's firmware which was not optimized for little feedback latency. As the feedback latency turned out to have a significant negative impact on input accuracy and speed when using tactile feedback instead of visual feedback, it was minimized from $> 120ms$ to around $30 - 50ms$ in several iterations by adapting configuration, optimizing software and replacing the standard vibration motors with the fastest responding tactile feedback devices found on the Internet which turned out to be linear resonant actuators by precisionmicrodrives[1] (still with a response+startup time of ~$20ms$). As of now the tactile feedback precision is comparable to the visual feedback precision. There is still room to lower feedback latency by using software prediction (described in the last chapter) but during tests with visual feedback the bottleneck was the subjects' motoric accuracy. A test matching the tactile feedback instead of the visual feedback against the subjects' motoric accuracy was not done with the latest optimized feedback latency. While it is assumed that the bottleneck will have moved towards the motoric accuracy, for trained, high performing typing that may change again though. Having a too large feedback latency impacts the design of the input system.

An unanticipated positive finding during intermediate tests was the subjects' accuracy

[1] https://www.precisionmicrodrives.com/product/c08-005-8mm-linear-resonant-actuator-3mm-type

of small finger movements. They could move their fingers precisely in up to $2° - 4°$ steps. As you can quickly check on your own hand, you can move each of your fingers in a range of approximately $90°$ at the joint connecting to your palm. The input system uses 6 zones per fingers that makes an operating angle of $12° - 24°$, leaving enough buffer so that issues because of a limited dexterity range were not confronted yet.

Chapter 5

Evaluation

There is no evidence yet that a glove based text input can match the performance of a keyboard or even the lower performance of a touchscreen keyboard in terms of WPM. This chapter describes the testing procedures and the test results of the last evaluation iteration cycle of this thesis, which tests intuitiveness, learnability and typing performance of the glove against the keyboard. As the evaluation doesn't include training, it tries to simulate a first use scenario.

5.1 Setup

The evaluation is led in German and split into two parts. In the first part the participants have to fulfill text typing tasks. For the typing task they use either the glove or the keyboard, either the english alphabet or a made-up glyph alphabet (Figure 5.1), either typing few letters (almost) without visual feedback or typing a longer text with time constraint. All text samples can be seen in Figure 5.2. The participants fulfill all $2^3 = 8$ configurations in randomized order to diminish the effect of inter-task learning effects on the analysis. During and between the tasks, the participants can trigger a help-screen which explains the usage of the glove input system. During evaluation, the glove's input alphabet is constrained to english characters, punctuation symbols and digits to reduce cognitive overhead.

The typing tasks which require the participant to omit visual feedback hide the visual feedback for the glove and a cardboard is put on top of the keyboard for the other case. The participant still can lookup the visual feedback if needed but s/he is told to not do

Figure 5.1: Glyphalphabet used during evaluation.

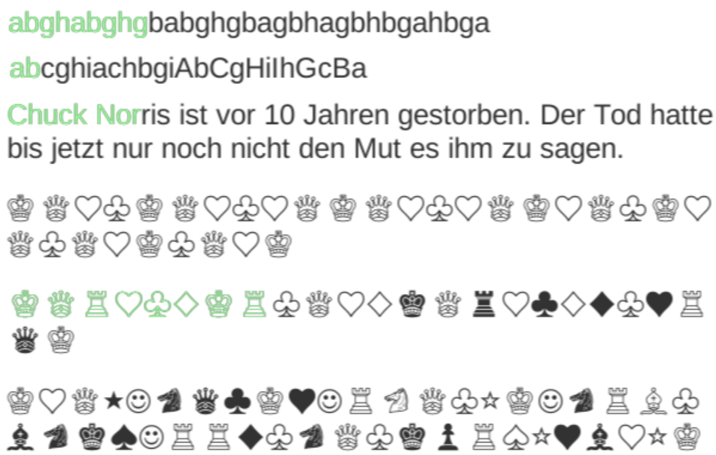


Figure 5.2: Textsamples used during evaluation.

it if at all possible and s/he has to press a key to do the lookup. Before the participant is told to write the text blindly, s/he first writes the same text with the visual feedback. The omission of the visual feedback enables comparison of the tactile feedback. There are only 12 distinct symbols used for the blind-writing tasks to decrease learning time for both, glove and keyboard. The time constraint for the longer text typing task is 1 minute.

Each input stroke is stored together with timestamp, current task, typed symbol and expected symbol. Furthermore each help-screen lookup and each visual feedback lookup for blind-writing tasks is stored together with time and current task.

Before the first task requiring glove usage the participants have the possibility to experiment and get acquainted with the input system and have to write one word as test before proceeding to the next task. The measures taken for analysis are:

- M1: characters per minute, excl. errors
- M2: character error rate
- M3: visual feedback lookups
- M4: help lookups

The second part of the evaluation is a questionnaire. The questions are:

- 1. "Bitte geben Sie Ihr Alter an." / age
- 2. "Bitte geben Sie Ihren Beruf an." / profession
- 3. "Bitte geben Sie Ihr Geschlecht an." / gender
- 4. "Wie oft verwenden Sie ein Keyboard?" / keyboard usage frequency
- 5. "Verwenden Sie 10-Finger-Schreiben?" / touch typing
- 6. "Haben Sie einen Datenhandschuh früher schon einmal ausprobiert?" / earlier data glove usage
- 7. "Wie wahrscheinlich schätzen Sie es ein, in den folgenden Szenarien den Datenhandschuh zum Schreiben zu verwenden?" / usage scenarios
- 8. "Welche Aspekte am Datenhandschuh haben Sie gestört?" / difficulties during usage

- 9. ”Welche Aspekte würden den Datenhandschuh für Sie verbessern?” / suggested improvements

- 10. ”Haben Sie sonstige Anmerkungen?” / other comments

The answers were linked to the results of the first part.

5.2 Results

The participants' age is 10-20 (1 subject), 20-30 (4 subjects), 30-40 (1 subject) and > 50 (2 subjects). Two are female, six are male. 4 of the 8 subjects already used a data glove before.

During the time-constrained tests, the participants reached 3WPM with the glove for both, glyph and english typing tasks (Figure 5.3). They reached a keyboard performance of 8WPM for glyph input and 6-100WPM for english text input. WPM is computed from the median time between two correct input strokes. 5 subjects have office jobs, 3 have non-office jobs. 1 subject uses the keyboard one time or less per week, 1 subject uses it less than one hour per day and 6 subjects use it more than one hour per day. 4 of the 8 subjects use touch typing.

4 of the 8 subjects used the help button and looked at the usage manual, the other 4 didn't have any explanation on the usage of the glove.

6 of 15 blind-typing tasks with a 12-element alphabet using the glove were typed with more than 10 symbols typed in sequence without looking at the visual feedback on average. That compares to 14 of 16 when using the keyboard (Figure 5.5). Most lookups were caused by the pinky finger unconsciously drifting out of its input zone.

Typing accuracy, measured as #correctInputs divided by #inputs, is in the range 55%-97% for the keyboard and 8%-42% for the glove (Figure 5.6).

Most usage scenarios were rated as very improbable. There was one participant rating with the requirement of an improved system, thus selecting more probable answers. Most probable usage was approved for a VR application, least probable usage for working use cases. Except for the VR application, all use cases were rated improbable (Figure 5.7).

In the free text answers and during observation, all participants mentioned strain in the pinky finger which needed to be stretched the whole time during glove typing. Some also mentioned strain in the ring finger. The participants had to position their hand into a position so that their fingers were freely movable. The USB-cable which is needed

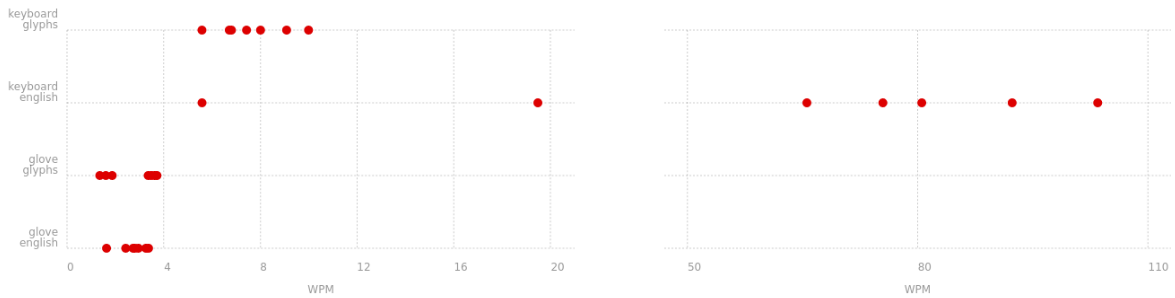


Figure 5.3: Writing performance during the time-limited typing tasks. The glove performance (third and fourth row) with around 3WPM compares to a keyboard performance of 8WPM for untrained glyph input and 6-100WPM for english text input.

by the current glove prototype constrained the number of possible body positions significantly. Furthermore moving fingers independently from each other was challenging for the subjects.

Some subjects wished stronger vibrational response and coarser input ranges. There was some minor confusion with the visual feedback when the hand was rotated with the inside up, as the visual columns didn't match the finger order anymore. There was also confusion with the auto-align feature that slowly aligns the center of a selected zone with the current finger position to avoid accidental drifting-out over a longer period. The feature causes changing absolute input positions which confused some participants.

Many subjects tried to prevent false input and had negative emotions when they knew they did a false input. Some had a competitive feeling during the typing tasks motivating them to type faster and increasing fun during evaluation. Some were frustrated by the demanding typing task, one subject canceled a task for this reason.

The light zone vibration patterns meant for absolute positioning were not distinguished between each other. Zone vibrations from different fingers were perceived as one single vibration, not knowing which finger was vibrating and which wasn't.

Many couldn't find the space symbol during glove typing and asked where to find it.

One participant had a rather small hand so that the sensor on the pinky finger didn't work flawlessly.

Figure 5.4: WPM performance for different task order. Whether the participant accomplished the english typing tasks before the glyph typing tasks or after them, seems to have no effect on the typing performance (red: glyph typing tasks before english typing tasks, blue: english typing tasks before glyph typing tasks). Probably the texts are to short to show a learning effect.

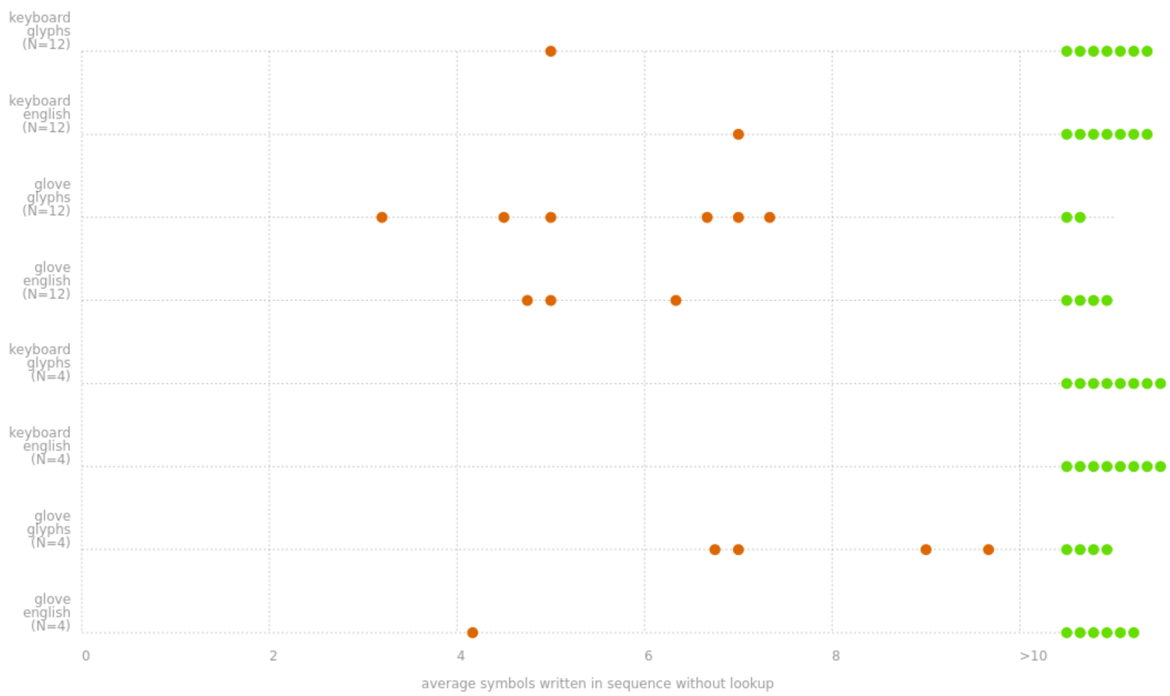


Figure 5.5: Average symbols written in sequence without lookup during blind-typing tasks. In the tasks with an alphabet size of 12, only few participants typing with the glove get along with avoiding the visual feedback. The green circles in the chart represent the participants that typed more than 10 sequential symbols without visual feedback on average.

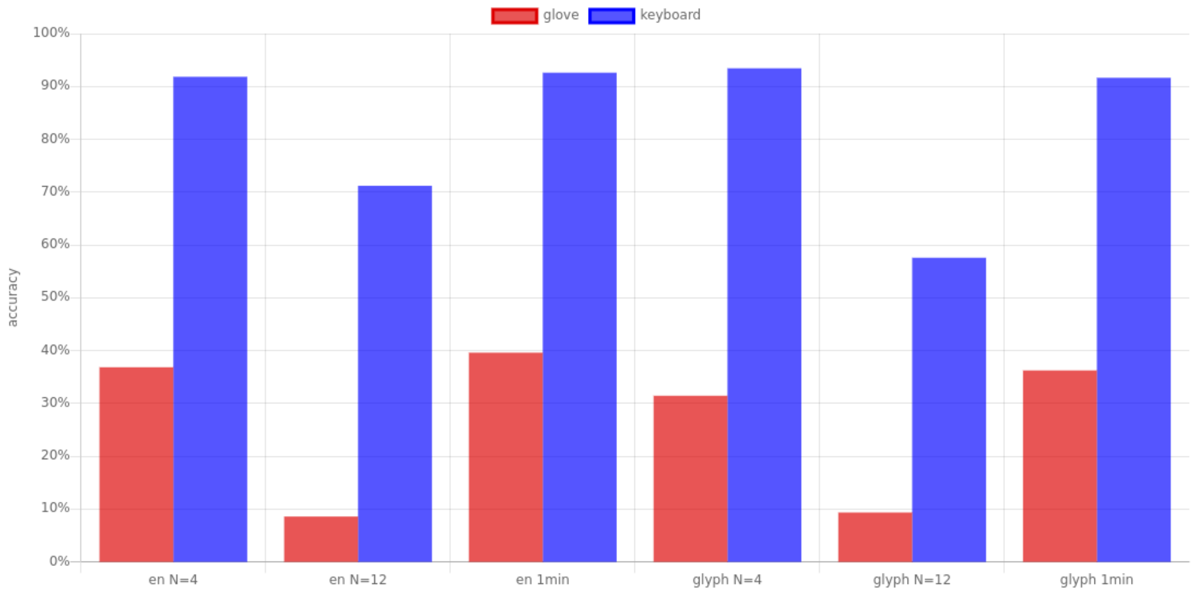


Figure 5.6: Input accuracy per test. Accuracy is decreased by inputting incorrect symbols.

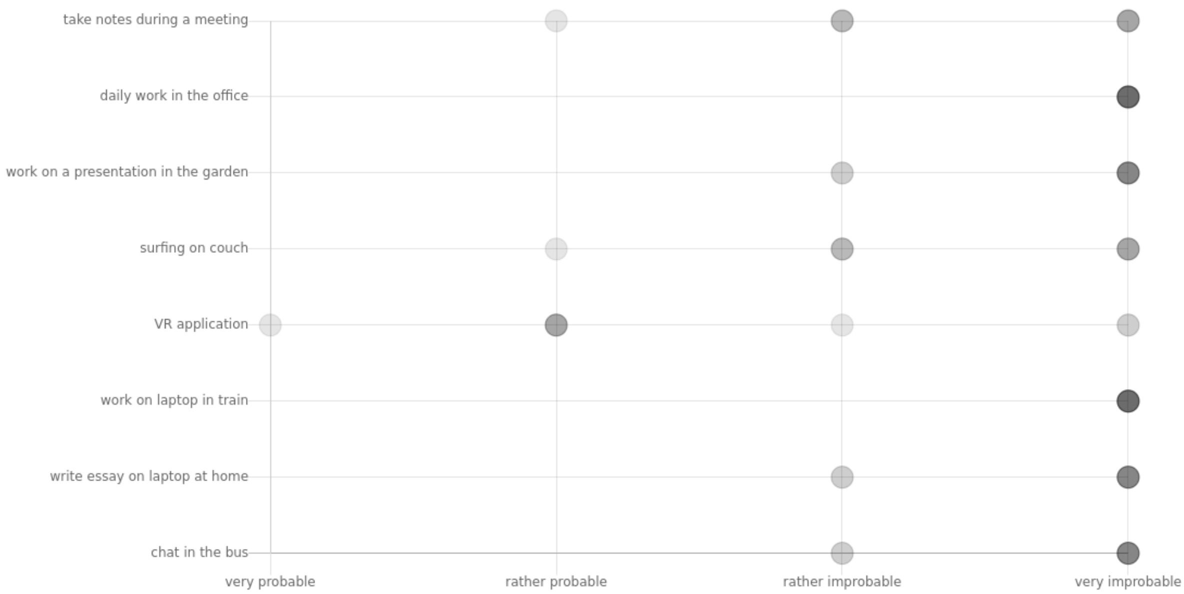


Figure 5.7: Usage scenarios. The person choosing the option "very probable" mentioned in the free text answers, that this is under the assumption of an improved system.

5.3 Analysis

Typical WPM rates for prose text typing can be seen at the website `typingtest.com` where everyone can do a self-test. The website claims the average typists to type 36 WPM while the average touch typist is claimed to type 58 WPM. The Guinness world record for typing a 160 character text on a touch keyboard is at 113WPM[1], for stenotyping it's 360 WPM with 97.23% accuracy[2]. In this thesis' evaluation the typing performance for typing english text on a keyboard reached around 100WPM. These speeds are only reached after a fair amount of training. When evaluating glove typing, providing an equivalent amount of training is very time-consuming. As the results show, using an untrained glyph alphabet on the keyboard drastically reduces typing speed, making a comparison with untrained glove typing more feasible. Still the movements needed to press keys on the keyboard are more familiar than the special finger gestures with the glove. A comparison with keyboard typing performance of people who never used a keyboard before would show whether this effect has significance. As it stands, the glove's typing performance didn't reach even half of the keyboard's typing performance. Also the typing accuracy is only a half to a third of the keyboard's value.

Non-visual typing, i.e., touch typing on the keyboard and vibration-based typing on the glove, shows a similar result. Touch typing on the keyboard worked for almost all participants while for the glove typing tasks more than half of the participants had to look up the visual feedback for less than 10 symbols in a row on average. The goal to enable non-visual typing is thereby not met. Though the concept that it can work was proven by the subjects who achieved typing more than 10 symbols in sequence without using the visual feedback.

The tension in the pinky finger is unacceptable and can be addressed either by a snapping feature or by substituting the pinky finger altogether. The mentioned tension in the ring finger should be addressed with a gesture system that doesn't require the finger to stay in one position for a long time.

The wish for a coarser input range must be probably accompanied with a more constrained alphabet to accommodate a shrunk input set. Further fingers and finer input gestures can be optionally added while advancing in training, similar to learning keyboard shortcuts after first using the keyboard without them. Stronger vibrations can be achieved by making the vibration intensity manually adjustable.

Although some subjects described the system as non-intuitively, the 4 subjects who used it without needing any instruction and without looking at the provided usage help show

[1] https://www.guinnessworldrecords.com/world-records/fastest-time-to-type-a-text-message-(sms)-on-a-touch-screen-mobile-phone

[2] https://www.guinnessworldrecords.com/world-records/fastest-realtime-court-reporter-(stenotype-writing)

that it can be comprehended just by the cues given by the visual and haptic feedback. While there is a lot of room to further enhance the visual cues, it already reached its goal in intuitiveness to be used without explanation.

Chapter 6

Conclusion And Future Work

The evaluation shows that the developed input system used in this work can't compete against the keyboard in performance and accuracy. This chapter will talk about how the results can be used and mention some steps that can further improve the implemented input system.

The electronic industry met the technical requirements to build a wearable, little-strain, tactile-based typing device for many years already. Furthermore many people already tried out the possibility to use a glove as typing device for tens of years yet it didn't get very popular. There may be people using them but they are a minority. This thesis doesn't have a superior setup and the software uses basic structures and algorithms. The hardware used in the glove is faster and more accurate than gloves in related work as their manufacturing improved over the years but that doesn't change anything conceptually.

This work didn't reach its goal to get keyboard typing performance parity with a more user-friendly device, the results are not convincing at all yet. But it is an insight in state-of-the-art data glove usage for typing and there is a lot of room for improving the implemented input system.

6.1 Further Development

Further development cycles are specified that were not processed due to organizational constraints. Iterations to increase WPM-performance are the evaluation of alternative gestures, a manual input trigger and having more-dimensional input per finger instead of one-dimensional by using the sidewards angle of the fingers and both bending angles instead of only one. Measuring the sidewards angle requires a more precise sensor calibration than is currently the case.

Furthermore the feedback latency can be reduced from currently ~$30-50ms$ to assumed $< 10ms$ via positional prediction implemented in software. This is possible because of the slow rate of 5-10Hz a human can control her/his fingers which is orders of magnitude slower than s/he can perceive tactile stimuli. Going down to $< 1ms$ still would be demanding with the used hardware though because of the vibrational feedback's startup-time and the limited processing power of the embedded chip.

Iterations to increase usability are the implementation of a 2d-pointing feature, standard USB-HID implementation, wireless USB plug, pressure-based input, a fallback system for hand-injured, coarser gestures for beginners, layout optimization and a richer visual feedback.

Further performance improvement can be achieved with symbol-prediction in the client application.

6.2 Follow Up

If you want to stay informed of whether the text input system with the Cynteract glove will sufficiently evolve to match or over-perform classic keyboard's typing speed, you can visit the website `cynteract.de` in several months/years or contact me at `adam.brunnmeier@rwth-aachen.de` .

Bibliography

[AS12] C Amma and T Schultz. Airwriting: Demonstrating mobile text input by 3d-space handwriting recognition. 2012.

[BSF14] R Barros, M Soares, and M Fernandes. Ergonomic evaluation of manual force levels of the elderly in the handling of products: An analysis using virtual reality. 2014.

[CD12] F Camastra and D DeFelice. Lvq-based hand gesture recognition using a data glove. 2012.

[CFH03] F Chen, C Fu, and C Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. 2003.

[CGM00] Y Chen, W Gao, and J Ma. Hand gesture recognition based on decision tree. 2000.

[CYKW18] T Chan, Y Yu, H Kam, and K Wong. Robust hand gesture input using computer vision, inertial measurement unit (imu) and flex sensors. 2018.

[EC14] F Erden and A Cetin. Hand gesture based remote control system using infrared sensors and a camera. 2014.

[FJSD13] H Fonseka, J Jonathan, P Sabeshan, and M Dissanayaka. two handed sign language recognition system using image processing. 2013.

[FR94] William Freeman and Michal Roth. Orientation histograms for hand gesture recognition. 1994.

[GSK12] A Gupta, V Sehrawat, and M Khosla. Fpga based real time human hand gesture recognition system. 2012.

[KPBB02] J Kim, K Park, W Bang, and Z Bien. Continuous gesture recognition system for korean sign language based on fuzzy logic and hidden markov model. 2002.

[KTG+04] T Kuroda, Y Tabata, A Goto, H Ikuta, and M Murakami. Consumer price data-glove for sign language recognition. 2004.

[LCoS11] S Lee, S Cheong, C ooi, and W Siew. Real time fpga implementation of hand gesture recognizer system. 2011.

[LM13] A Linsie and J Mangaiyarkarasi. hand gesture recognition using mems for specially challenged people. 2013.

[LO98] R Liang and M Ouhyoung. A real-time continuous gesture recognition system for sign language. 1998.

[LP14] O Luzanin and M Plancak. Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. 2014.

[LW03] M Lee and W Woo. Arkb: 3d vision-based augmented reality keyboard. 2003.

[MT07] K Murakami and H Taguchi. Gesture recognition using recurrent neural networks. 2007.

[PFN+08] V Pamplona, L Fernandes, L Nedel, J Prauchner, and M Oliveira. The image-based data glove. 2008.

[SAP16] S Shabanabegum, C Amutha, and V Priya. Implementation of hand gesture recognition system in fpga using pulse-coupled neural network. 2016.

[SAW+14] M Saad, M Ali, K Wan, D Hazry, S Ab, and Z Razlan. Experimental studies of the correlation between fingers bending angle with voltage outputted from glovemap. 2014.

[Smi13] J Smith. Gesture recognition using neural networks. 2013.

[SNSK18] B Sudha, V Naik, S Sarveshwar, and V Karimallappanavara. Data-glove as a sign recognition system. 2018.

[TKO10] Y Tabata, T Kuroda, and K Okamoto. Basic study toward automatic generation of glove-type command input device with optimal number of sensors. 2010.

[TL03] K Tarchanidis and J Lygouras. Data glove with a force sensor. 2003.

[TSMM17] D Tootaghaj, A Sampson, T Mytkowicz, and K McKinley. High five: Improving gesture recognition by embracing uncertainty. 2017.

[WJ07] H Witt and T Janssen. Comparing two methods for gesture based short text input using chording. 2007.

[WP14] D Way and J Paradiso. A usability user study concerning free-hand microgesture and wrist-worn sensors. 2014.

[WS99] J Weissmann and R Salomon. Gesture recognition for virtual reality applications using data gloves and neural networks. 1999.

[XZL+18] R Xu, L Zhai, J Li, J Jiang, W Zhai, and H Yu. New application of children's sign language enlightenment based on data glove. 2018.

[YYQ+18] S Yin, J Yang, Y Qu, W Liu, Y Guo, H Liu, and D Wei. Research on gesture recognition technology of data glove based on joint algorithm. 2018.