RWTH AACHEN UNIVERSITY

OpenVNAVI

A Vibrotactile Navigation Aid for the Visually Impaired

Bachelor's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
David Antón Sánchez

Thesis advisor:
Prof. Dr. Jan Borchers

Registration date: 04/05/2015
Submission date: 21/08/2015

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, August 2015
David Antón Sánchez

Contents

List of Figures

List of Tables

Abstract

According to the World Health Organization 285 million people worldwide are estimated to have a degree of visual impairment, with 39 million of them suffering from complete vision loss.

The current degree of technological development of many fields has brought and continues to bring comforts and life-changing improvements to our lives: smartphones, GPS, self-driving cars, etc. However, blind and visually impaired (BVI) people still rely on century-old methods for navigating the world on their own:

The white cane, a simple, affordable tool used by the BVI community as an obstacle detection device by means of direct interaction with the obstacles; and the guide dog, used as an obstacle avoidance and navigation aid, although not affordable for the vast majority of the BVI community.

The aim of this Bachelor Thesis is to create a low-cost system based on vibrotactile feedback that improves upon the functionality of the guide dog as an obstacle avoidance and a navigation aid, allowing more BVI users to navigate the world easily and safely.

First, an exploration of the related work from the past decades is performed, analyzing the state of the art and the current areas of research. Then, the author analyzes the requirements and describes the implementation of all the features and components of the system. After that, an evaluation of the implementation of the system is performed. Finally, an exploration of the possible future development is presented.

Überblick

Nach Schätzungen der Weltgesundheitsorganisation leben weltweit ca. 285 Millionen Menschen mit einer Sehbehinderung, wovon 39 Millionen komplett erblindet sind. Der technische Fortschritt hat unseren Alltag inzwischen durch Smartphones, GPS etc. in vielen Bereichen verändert und erleichtert und alles deutet darauf hin, dass Entwicklungen wie beispielsweise selbstfahrende Kraftfahrzeuge unser Leben zukünftig noch bequemer und komfortabler machen werden. Demgegenüber stehen blinden und sehbehinderten Menschen immer noch hauptsächlich jahrhundertalte Methoden zur Verfügung, um sich eigenständig in ihrer Umgebung zu orientieren:

Der weit verbreitete Blindenstock, ein einfaches, preisgünstiges Hilfsmittel, das zum direkten Erkennen von Hindernis benutzt wird; sowie der Blindenhund, welcher zum Vermeiden von Hindernissen und als Navigationshilfe eingesetzt wird, aufgrund hoher Kosten aber für viele Betroffene nicht infrage kommt.

Das Ziel dieser Bachelorarbeit ist die Entwicklung eines preisgünstigen Hilfsmittels, welches durch vibrotaktiles Feedback zur Vermeidung von Hindernissen und als Navigationshilfe eingesetzt werden kann. Es soll blinden und sehbehinderten Menschen erleichtern sich einfach und problemlos in ihrer Umgebung zurechtzufinden.

Im Folgenden werden der aktuelle Forschungsstand auf diesem Gebiet dargelegt und bisherige Entwicklungen analysiert. Anschließend beschreibt der Autor die Anforderungen an das in dieser Arbeit entwickelte System, dessen Eigenschaften und einzelne Komponenten, sowie die schlussendliche Umsetzung. Hierauf folgt eine Beurteilung der Implementierung dieses Systems. Abschließend werden zukünftige Entwicklungsmöglichkeiten aufgezeigt.

Acknowledgements

First, I would like to thank Dipl.-Inform. René Bohne for supervising this bachelor thesis, Jan Thar for his valuable tips and Prof. Dr. Jan Borchers for his help and for giving me the wonderful opportunity of making this project at the Media Computing Group. Without the Fab Lab, OpenVNAVI would not be a reality.

I thank my parents for all the support they have given me over the years and for giving me the chance to pursue my dreams. I thank all my friends, especially Mario, Juan, Carlos, Elena and Alberto for their understanding for all the time I could not spend with them during long study hours. I am also grateful to Kirstin for all the things we shared and for making me spend a wonderful year in Germany.

And last but not least, I would like to thank the professors that inspired me, especially Juan and Alberto, for making me discover the world of 3D printing and the open source culture; and my friend and colleague David, always a source of inspiration, for constantly making me strive for improvement as an engineer.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Source code and implementation symbols are written in typewriter-style text.

```
myClass
```

The whole thesis is written in Canadian English.

Download links are set off in coloured boxes.

File: myFile[a]

[a] http://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

In this chapter, the author offers an insight into the current mobility aids for the blind and visually impaired and the motivation behind OpenVNAVI, as well as a description of the different chapters that compose this Bachelor Thesis.

1.1 Background

Blindness may result from a disease, injury or other conditions that limit vision. According to the World Health Organization 285 million people worldwide are estimated to have a degree of visual impairment, with 39 million of them suffering from complete vision loss.

Unfamiliar environments may pose challenges for blind people with complete blindness. Often they even find it difficult to navigate well-known environments mainly due to moving obstacles like those found on a crowded street. This fact imposes a limit on their ability to independently navigate the environment forcing some of them to bring a sighted friend or family member to guide them [WHO].

The most popular tools in the blind and visually impaired (BVI) community are the white cane and the guide dog [WHO].

1.1.1 White Cane

The white cane is a a simple, affordable tool used by the BVI community as an obstacle detection device. It has been the status quo for over a century, remaining as the most popular tool in the BVI community. Its main purpose is obstacle detection, which means that users have to physically make contact with obstacles with the white cane to be able to detect them.

The white cane cannot detect overhead obstacles

However, the white cane is not able to provide additional information such as the speed and type of object the user is encountering and the static or dynamic nature of the obstacle. Although it is popular and affordable there still exist a 40% chance of missing potholes, sidewalk cracks and curbs and also causes constant joint pain due to the swinging action.

Figure 1.1: Blind person using a white cane [Shutterstock]

1.1.2 Guide Dog

The main purpose of guide dogs is to help BVI users navigate the environment by helping them avoid obstacles in their path and prevent dangerous situations. Guide dogs can only learn a limited number of routes and thus cannot be used effectively as a standalone navigation device, forcing BVI users to learn their current position and destination and notifying the dog when they want to change course.

Guide dogs are not affordable for the majority of BVI users

The special bond that BVI users have with their guide dogs increases their confidence and reduces their stress levels as they provide support, security and companionship [Whitmarsh, 2005]. But guide dogs are not affordable for the vast majority of the BVI community, with costs to support them from birth to retirement reaching £50.000 in the U.K. [Guide Dogs UK, 2015]

Figure 1.2: Blind person with a guide dog [Agencia Brasil]

1.1.3 Other Mobility Aids

Some manufacturers offer canes with extended functionality like the UltraCane, the BAT K-Sonar or the Sonic Pathfinder, but they never enjoyed popularity due to the

GPS-based outdoor navigation with smartphones is becoming more popular

limited benefits of using them compared to the status quo and multiple other factors, including cost, usability, and performance. There exist many different devices on the market that offer GPS-based outdoor navigation. In the recent years the popularity of this kind of Electronic Travel Aids (ETAs) has increased thanks to smartphone apps that offer this functionality at a reduced cost.

A survey for the 2007 Annual Report from the American Printing House for the Blind found that only 10% of U.S. children use Braille as their primary reading medium. Text-to-Speech seems to be more convenient for most applications when most BVI users, for instance, are not able to read, or even find braille information on signs placed on the wall. Efforts to integrate Braille information on non-standardized locations are most of the times futile, urging researchers to come up with new ways to convey information more efficiently without the need of complex and costly deployments.

Blind people can still suffer accidents while using current navigation aids

Neither the long cane nor the dog guide can protect the user from all types of hazards. Obstacles at head height are beyond the volume surveyed by the cane and thus impossible to detect. In a recent survey of 300 blind persons, 13% of them reported that they experience head-level accidents at least once a month [Manduchi, 2011].

There is also still room for improvement and issues that need to be addressed to improve the current systems such as indoors navigation and wearable obstacle avoidance devices that can provide BVI users with enough confidence and safety that would enable them to navigate the world like the rest of the population.

1.2 Motivation

The current degree of technological development of many fields has brought and continues to bring comforts and life-changing improvements to our lives: smartphones, GPS, self-driving cars, etc. However, BVI users still rely on century-old methods for navigating the world.

The aim of this Bachelor Thesis is to create a low-cost navigation system based on vibrotactile feedback that improves upon the functionality of the guide dog as an obstacle avoidance and a navigation aid, allowing more BVI users to navigate the world easily and safely.

The flexibility of this system should allow for the integration of different features that the current status quo cannot provide and it should strive for offering enough tangible benefits to using them to make BVI users take a leap into a new generation navigation aids worthy of the 21st century.

1.3 Thesis Overview

In chapter 2 "Related work", an exploration of the related work from the past decades is performed, analyzing the state of the art of navigation aids for BVI users and the current areas of research. In chapter 3 "Own work", the author analyzes the requirements, explores the technologies available on the market and describes the implementation of all the features and components of the system.

In chapter 4 "Evaluation", an evaluation of the performance of the different components of the system is performed. Finally, in chapter 5 "Summary and Future Work", the author presents an exploration of the possible future development and improvements.

OpenVNAVI is open source software and hardware. The code and the design files generated by the author and detailed documentation are publicly available under a GPLv3 license and hosted on GitHub.

OpenVNAVI GitHub Repository[a]

[a] https://github.com/davidanton/OpenVNAVI

Chapter 2

Related work

In this chapter the author explores the main wearable Electronic Travel Aids (ETAs) based on sensory substitution and analyzes the state of the art of ETAs based on tactile-visual substitution.

2.1 ETAs Based on Sensory Substitution

As of 2015 and despite intensive research by many groups all over the world over the past decades, the issue of delivering a new, effective navigation and collision avoidance system for BVI users remains still unsolved.

ETAs based on sensory substitution share the same operation principle: scanning the environment and displaying that information through another human sense, generally hearing and touch. ETAs fall into two categories: portable and wearable. Portable devices are usually lightweight and can be easily carried by the user, requiring constant interaction, whereas wearable devices are worn on the body and allow for hands-free interaction. The area of research related to wearables for assisting people with disabilities is still young and experimental, with not many mature commercial products on the market.

The sense of hearing is the most important sense for blind people

For BVI people, hearing and touch are their first and second main human senses, respectively. They rely on hearing environmental cues for awareness, orientation, mobility and safety. Given the importance of the sense of hearing for blind people, any impediment that lowers its performance will pose an additional problem to them as it might interfere with their ability to gather environmental cues [Álvarez Sánchez et al., 2005].

Continuous acoustic feedback is detrimental to their well-being

Humans have a limited capacity to receive and process information taken from the environment and too much information from one sensor modality can cause sensory overload. Continuous acoustic feedback could affect their stress levels, posture and equilibrium [Hakkinen et al., 2002].

Tactile-visual substitution demonstrated better performance than Acoustic-visual substitution

A survey of current wearable systems conducted by Tapu et al. [2014] that included devices performing tactile-auditory and tactile-visual sensory substitution concluded that the Tactile Vision System (TVS) from Johnson and Higgins [2006a] and, the EPFL Project from Cardin et al. [2007], both wearable, tactile-visual obstacle avoidance ETAs that map depth information into vibration, delivered the best overall performance according to their metrics. Therefore, in this study the author will focus only on wearable tactile-visual substitution ETAs.

Figure 2.1: Tactile Vision Substitution (TVS) system. [Johnson and Higgins, 2006b]

2.1.1 Tactile-visual Sensory Substitution

Tactile-visual sensory substitution has been researched since the 1960s: [Geldard, 1960], [Geldard and Sherrick, 1965], [Lechelt, 1985], [Velazquez et al., 2005], [White, 1970], [Bliss et al., 1970], [Bach-y Rita, 2004]. Tactile feedback is a viable choice for ETAs, however, the information presented must be adapted in accordance with the area of the body where the information is displayed. ETAs based on tactile-visual sensory substitution can be placed on multiple different areas of the human body such as the hand, wrist, feet, chest, abdomen or even the head, tongue, or ears.

The current level of miniaturization of electronics and actuators allows for the creation of devices that can be integrated into clothing, allowing BVI users to perform daily tasks without feeling bothered by them.

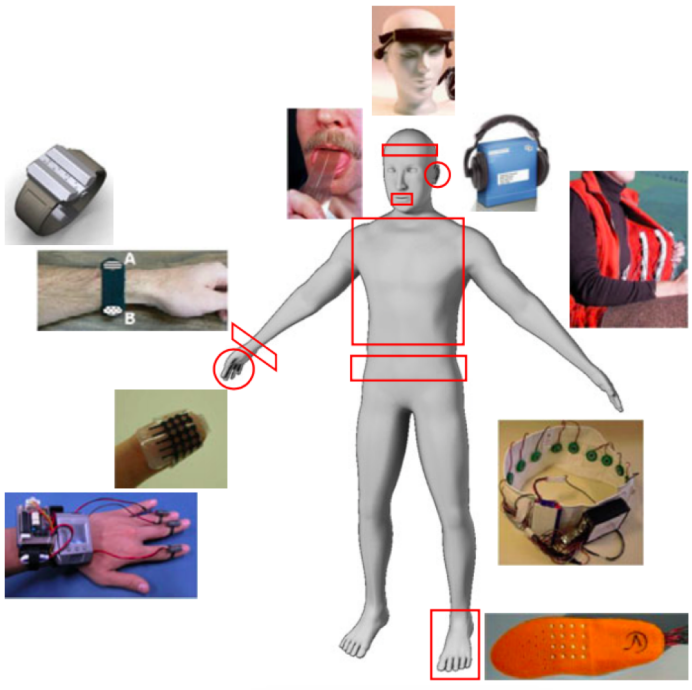


Figure 2.2: Overview of tactile-visual wearable assistive devices for the blind [Velázquez, 2010]

Many variables affect the perception of vibratory stimuli, including vibration amplitude, frequency, duration, contact geometry, contact area, contact force, context, temperature, age and pathology [Velázquez, 1999]. Regarding pathology, it has ben shown that with respect to healthy individuals, the tactile acuity of blind individuals is enhanced [Kay, 2000].

Assistive devices worn on the fingers like one from Koo et al. [2008] are able to provide limited Braille information or vibration alerting. Devices worn on the wrist were only able to provide simple vibration alerting [Ng et al., 2007]. Assistive devices worn on the hand are able to provide more complex information [Shrewsbury, 2011]. However, by wearing them on the hand as gloves, the hands are always busy and the mapping of the depth information changes with the position of the hands relative to the torso, making it confusing. Assistive devices worn on the feet can deliver simple information and vibration alerting patterns, however complex patterns or shapes cannot be accurately perceived [Velazquez et al., 2009].

Wearable electronic travel aids should not be invasive or obtrusive

In Mann et al. [2011], researchers used a depth sensor mounted on a helmet with a onedimensaional array of vibration motors on the forehead, proposing an integration into a device the size of a regular pair of glasses. Kaczmarek [2011] developed the tongue display unit (TDU) in 1998 at TDU Wisconsin, a device that used 144 microelectrodes to transmit depth information to the brain via the nerve endings of the tongue. Experiments show that after enough training the user loses awareness of the device and perceives the stimulation as shapes and features in space. However, the device has some disadvantages like the fact that the device has to be in the mouth, which makes it relatively invasive, and the limited ability of the tongue to discriminate information.

2.2 Wearable ETAs Worn on the Abdomen

The brain references direction mainly with the torso, which is relatively stable with respect to the path followed while walking [Johnson and Higgins, 2006a]. The belly area is ideal for tactile-visual substitution ETAs for several reasons: first, that area of the skin has the advantage of being relatively flat, large, and easily accessible compared to other areas of the body, in addition to being minimally used during navigation. Furthermore, the skin is capable of representing information in 2D and integrating signals over time like the retina [Geldard, 1960], meaning that visual patterns and tactual patterns are functionally interchangeable [Kaczmarek and Bach-Y-Rita, 1995]

The belly area has great potential for tactile-visual substitution ETAs

2.2.1 Devices Using 1D Actuator Arrays

Some research projects like Gemperle et al. [2001], Tsukada and Yasumura [2004] or Nagel et al. [2005] used belts with one-dimensional arrays of vibration actuators to deliver turn-by-turn navigation with vibrotactile patterns. Others used the same kind of actuator array but focused on obstacle avoidance, most of them with the help of ultrasonic sensors to obtain depth information.

The most relevant of these systems is the Tactile Vision Substitution (TVS) system by Johnson and Higgins [2006b]. The TVS is a compact wearable system that uses a stereo camera positioned on the belly to obtain depth information from the environment and represents it by applying different levels of vibration on a one-dimensional array of 14 vibration motors on a belt.

The system works in real time and provides the user with hands-free obstacle avoidance without blocking the sense of hearing. However, due to an insufficient number of vibration actuators and the fact that the information is rendered in one dimension, these systems fail to represent overhead obstacles and cannot represent complex information from a normal environment.

Onedimensional actuator arrays cannot accurately represent complex information

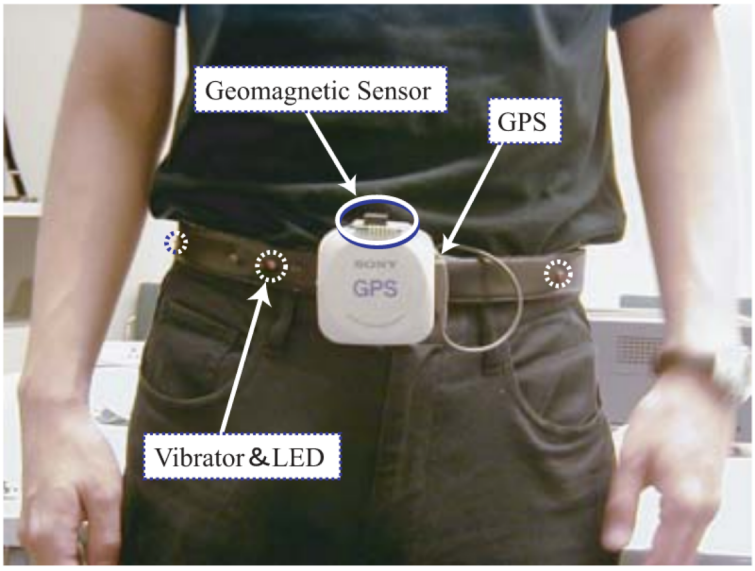


Figure 2.3: ActiveBelt system. [Tsukada and Yasumura, 2004]

Other academic projects that used 1D actuator arrays include Johnson and Higgins [2006a] and Cardin et al. [2007]. Some non-academic projects like Kinecthesia, Kinect for the Blind and viSparsh are also relevant.

2.2.2 Devices Using 2D Actuators Arrays

2D actuator arrays were tested in the 1970s for the first time

White et al. [1970] pioneered this kind of devices in 1970 with his TVS system. The device was composed of a chair with a 20x20 tactor array and a video camera. Visual information was captured using a video camera and delivered via the tactor array to the skin of the back. With the TVS system, users were able to perform eye-hand coordination and simple object recognition tasks. However, since the information was captured using a video camera and not a depth sensor, the stimuli became unclear in cluttered environments. Due to the size of the setup, the TVS cannot be considered a wearable device.

One of the most relevant systems from the past decade is the Tyflos system from Dakopoulos et al. [2007], using a 4x4 array of vibration motors on the belly area. The Usability Engineering group at TNO Human Factors, an applied scientific research institute in the Netherlands developed a tactile display with 128 vibrating elements attached to the interior of a vest with the purpose of conveying flight information to pilots [van Veen and van Erp., 2003].

By means of a high number of actuators on the belly, a device like this one could deliver a much richer tactile representation of the environment compared to the existing wearable ETAs developed in the recent years.

Figure 2.4: Tactile Torso Display. [van Veen and van Erp., 2003]

Similar examples of this kind of devices can be found in the literature: [Saunders, 1983], [Reyes-Ayala et al., 2001], [Pineda Garcia et al., 2013], [Wu et al., 2011], [Jones et al., 2006]. Other commercial or non-academic projects that used depth information from a sensor and rendered a 2D visual representation using tacitle-visual substitution include Eyeronman and 3D Haptic Vest by Sean Benson.

By focusing on obstacle avoidance rather than object recognition, the amount of information to be conveyed can be reduced to manageable proportions, making these systems very well suited for that purpose.

User acceptance of new assistive devices is still low

Despite all the efforts made by the research community, user acceptance of wearable assistive devices different from the status quo is still fairly low and remains as a challenge for the BVI population. However, the experience gathered over the decades has shown that creating successful assistive technologies is a difficult task. Cooperation between engineers, BVI institutions and users, along with great optimism and willingness to adapt to new paradigms are crucial factor for the development of these systems.

Chapter 3

Own work

In this section the author offers a detailed description of the requirements of the individual components that compose OpenVNAVI, as well as the procedures followed for the development of the OpenVNAVI prototype.

3.1 System Description

The depth sensor, positioned onto the user's chest scans the environment as the user moves. From the video feed of the depth sensor a frame is captured and then processed by the computer. Each frame is downsampled from 640x480 to 16x8 and each pixel is then mapped to a vibration motor unit forming an array positioned onto the user's belly.

The grayscale value of each pixel on the lower resolution frame is assigned to a PWM voltage value generated by the computer via PWM drivers that will drive each vibration motor obtaining a vibration amplitude value as a function of the proximity of an object.

PWM or Pulse-width modulation is used to control the power supplied to an electronic device

With this method the vibration motor unit array is able to represent a vibratory image onto the user's belly to help create a mental representation of the obstacles in the scene.

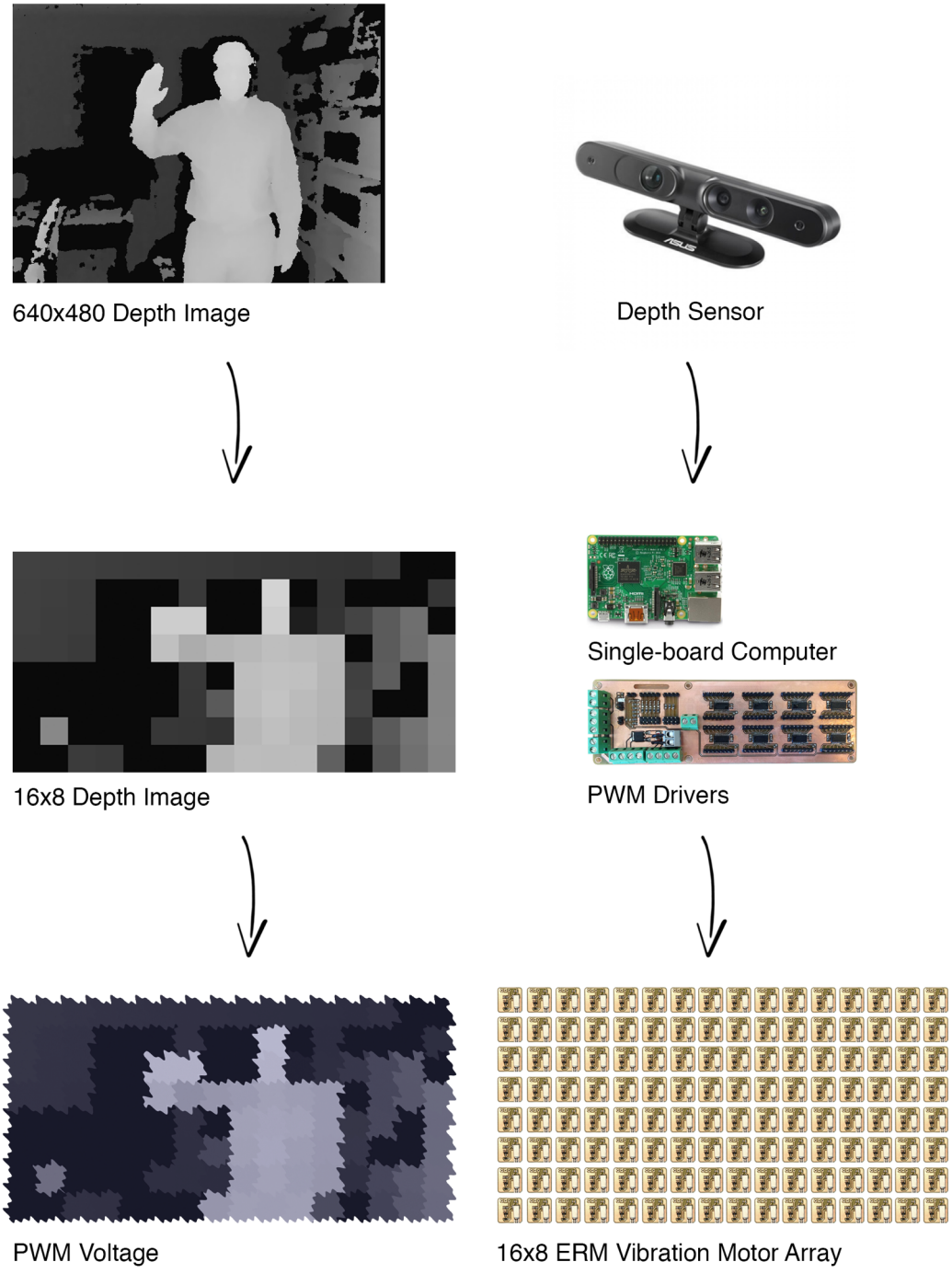


Figure 3.1: System overview

3.2 Requirements

Although the architecture of the system, as discussed in chapter 5 "Summary and Future Work", allows for the implementation of many other features, due to time constraints for the completion of this Bachelor Thesis the main goal of this system would be obstacle avoidance.

The prototype should look like a commercial product

OpenVNAVI should be as close as possible to a commercial product, from both an aesthetic and functional point of view. Social acceptance is an important factor to consider in any disadvantaged community, consequently wearing a device that can integrate with their clothes without drawing unwanted attention is always desirable. It should be as small and inconspicuous as possible allowing users with different body measurements to wear it comfortably.

An open-source license provides the rights to study, change, and distribute the software or hardware to anyone and for any purpose

As the name implies, OpenVNAVI should be open source software and hardware. For PCB and 3D design, KiCad and FreeCAD respectively are the best pieces of open source software available, making them the right choice for this project. The code and the design files generated by the author should be released publicly under a GPLv3 license. The use of custom electronics should also be favored over more expensive commercial alternatives. The full extent of the project should use as many pieces of open source software and hardware as possible, as well as off-the-shelf, easily accessible components, optimizing for cost.

3.3 Architecture

The technical requirements of each of the components of OpenVNAVI are here described in detail.

3.3.1 Depth Sensor

The depth sensor should have the same range as the white cane

The range of the depth sensor should span from just a few centimeters to at least 3 meters, allowing users to anticipate obstacles and act accordingly.

There exist different technologies to obtain 3D depth information and different manufacturers offer sensors with different technologies and capabilities. Some devices are capable of providing both RGB and depth information (RGB-D) whereas other only offer depth sensing capabilities. Hereunder an overview of the main technologies is described.

Triangulation (Stereo Vision)

Depth sensing using triangulation requires low-cost hardware

These systems use two cameras positioned at a fixed distance from each other. As is the case with the position of the human eyes, that distance produces a measurable disparity from which depth information can be extracted. Factors like color and texture of the objects and the illumination of the scene play an important role in the quality of the results.

Because the majority of the systems of this kind consist of just two cameras, one major drawback is the need to perform computationally expensive steps to obtain the depth information. To address this issue, some devices include an FPGA that performs the calculations, at the expense of a higher price point.

Structured light

Structured light sensors are sensitive to interferences

System of this kind work by projecting a known light pattern onto the scene, usually a pseudo-random pattern of dots, and then analyzing the distortion of that pattern caused by the objects in the scene, resulting in the reconstruction of the depth information. This technology has its limitation when it comes to measuring depth of objects with sharp angles with respect to the normal of the sensor and are also sensitive to optical interference caused by near-infrared light sources, such as sunlight. Due to this fact, this kind of sensors don't work very well outdoors. Another important factor to consider when analyzing this technology is the fact that when two or more devices are working side by side, the projected pattern of dots of each sensor will make them interfere with each other.

Time-of-flight (ToF)

Time-of-flight sensors offer better performance at the expense of a higher cost

The principle of operation of ToF sensors is based on phase shift measurements. A near-infrared LED illuminates a scene and the distance information is extracted by measuring the phase shift between the illumination and the reflection on the surface of objects. With this technology there exist a reduced dependence on scene illumination and almost no dependence on surface texturing, giving it an advantage in terms of quality of the results compared to structured light sensors. Due to the unique characteristics of this kind of sensor, manufacturing costs are higher than those of Structured light or conventional stereo vision. [Texas Instruments, 2014]

Analysis

The choice of a sensor for this project is affected by its measurement capabilities, performance, compatibility, ease of integration and cost. First, a feature comparison of the three main technologies is performed:

	Triang.	Str. light	ToF
Code complexity	High	Low	Low
Cost	Low	Medium	Medium
Compactness	Low	High	Low
Response time	Medium	Slow	Fast
Depth accuracy	Low	High	Medium
Low-light perf.	Weak	Good	Good
Bright-light perf.	Good	Weak	Good

Table 3.1: Comparison of different technologies [Texas Instruments, 2014]

Since OpenVNAVI needs to be mobile and the processing power of mobile single-board computers is limited, the use of conventional triangulation techniques with a stereo camera is discouraged.

Price is a very important decision criterion

Outdoor performance of structured light sensors and interferences with sensors of the same technology are factors that would affect the performance of a commercial product. Since low-cost sensors are based on this technology and further experiments will be carried out indoors, structured light sensors will be considered as a viable option for this prototype.

Based on the other two technologies available and the commercially available sensors on the market, the following analysis is performed:

	Kinect	Xtion PRO LIVE	Kinect 2	DepthSense
Technology	Str. light	Str. light	ToF	ToF
Depth resolution	320x240	320x240	512x424	320x240
Compactness	Low	Medium	Low	Medium
Range (m)	0.5 - 4	0.5 - 4	0.5 - 4	0.70 - 4 (@6fps)
Power supply	Yes (12V)	No (USB)	Yes (12V)	No (USB)
USB standard	2.0	2.0	3.0	2.0
RGB camera	Yes	Yes	Yes	No
OpenCV compat.	Yes	Yes	Yes	Yes
OpenNI compat.	Yes	Yes	Yes	No
Price (€)	170	139	184	234

Table 3.2: Comparison of different commercial sensors

The depth error of sensors based on PrimeSense technology (Kinect and ASUS Xtion PRO LIVE) is approximately 1 cm for objects at 2.5 m from the sensor [Khoshelham and Elberink, 2012] and increases proportionally to the squared measuring distance [Breuer et al., 2014]. Since precision is not considered a factor of critical importance for OpenVNAVI, the performance of all these sensors would meet the necessary requirements.

There exist no sensor that can span the full range of the white cane

It seems not to be possible for long range sensors to accurately perceive depth information for distances smaller than 50 cm. On the other hand, short range sensors can accurately provide depth data from around 10 cm but the maximum range is usually smaller than 1.5 m, making them not very good candidates for OpenVNAVI. This lack of sensing capabilities for distances shorter than 50 cm for long-range sensors could either be addressed by software or by attaching one or two ultrasonic sensors to warn about an impending collision.

The most popular device on the market is the Microsoft Kinect. Its low price point compared to other sensors at the time and the availability of drivers and tools for different platforms made it the sensor of choice for makers and

visual artists. Another device based on PrimeSense technology is the ASUS Xtion PRO LIVE. The main difference between this device and the Microsoft Kinect is mainly its compactness, being the former greater than the latter.

Community usage and software compatibility are two very important criteria

Taking into account the previous analysis of the current sensors available on the market and optimizing for cost, it can be concluded that the best option for this project is the ASUS Xtion PRO LIVE. This sensor delivers very good depth data, it is well known by the community, it is compatible with OpenNI and Open CV and also has a lower price point than its competitors.

Figure 3.2: ASUS Xtion PRO LIVE

3.3.2 SBC

The Raspberry Pi 2 is the most popular single-board computer

In order to meet the requirements of portability and processing power to process the data gathered by the sensor, the best option for this project is to use a Single-board Computer (SBC). SBCs are getting faster and cheaper every year, with Raspberry Pi leading the market.

In the following table, an analysis of 3 of the most popular SBCs is performed:

	Raspberry Pi 2	BeagleBone Black	MinnowBoard MAX
Compactness	High	High	High
CPU	ARM Cortex-A7	ARM Cortex-A8	Intel Atom
Architecture	ARMv7-A	ARMv7-A	x86-64
Clock speed	900 MHz	1GHz	1.33 GHz
Cores	4	4	2
RAM	1 GB	512 MB	1GB
Linux	Yes	Yes	Yes
I2C	Yes	Yes	Yes
GPIO	8-pin	65-pin	8-pin
Price (€)	39	63	184

Table 3.3: Comparison of different commpercial sensors

By analyzing the information from the table it can be concluded that the characteristics of those SBCs are very similar, being price the main differentiating factor. However, the Raspberry Pi 2 has a big advantage against its competitors. Not only does this device offer great technical specifications for an unbeatable price, but it also has great community support due to its popularity.

The software must be compatible with the hardware architecture of the single-board computer

Seemingly unimportant details like the type of ARM architecture can affect the compatibility with unsupported tools like OpenNI 1.x and they were taken into account when deciding which board to use. The fact that both hardware and software compatibility of the ASUS Xtion PRO LIVE sensor with Raspberry Pi 2 was successfully tested by the community providing good results, as well as its low price point and popularity, made the Raspberry Pi 2 the best option for the project.

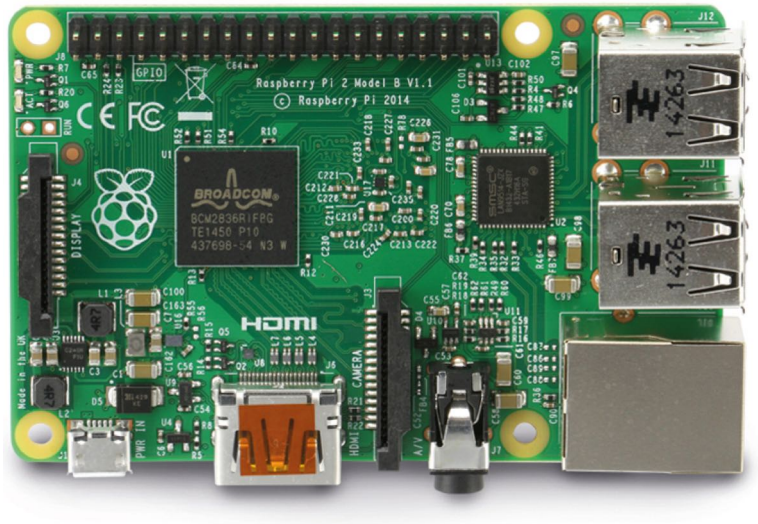


Figure 3.3: Raspberry Pi 2

3.3.3 Actuators

Vibration is a well-established, cheap and stable method of mechanical stimulation for providing haptic feedback. For this reason, haptic feedback based on vibration motors should be the technology of choice.

Humans perceive vibration through mechanoreceptors in our skin. Two kinds of mechanoreceptors, the Meissner and Pacinian corpuscles, detect rapid vibratory pressure and touch, with optimum sensitivities at 50 Hz and 250 Hz respectively [Kandel et al., 2013]. In order to optimize the performance of the actuators, the frequency of mechanical vibration should lie below 250 Hz.

Sensitivity decreases for vibration frequencies greater than 250 Hz

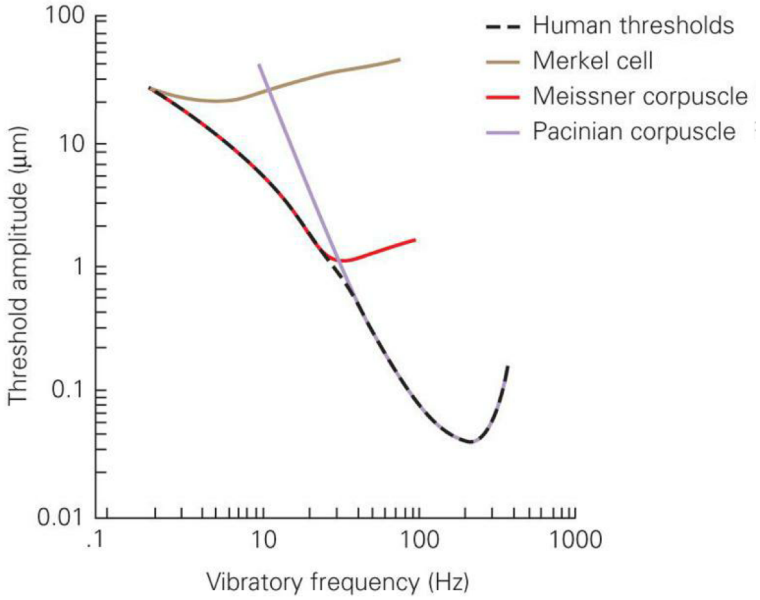


Figure 3.4: Sensitivity threshold [Kandel et al., 2013]

Actuators placed at a distance lower than the threshold would not be differenciated

The spatial resolution of stimuli on the human skin differs throughout the body. The two-point threshold measures the minimum distance at which two stimuli are resolved as distinct. If the separation is smaller, the stimuli are blurred into a single continuous sensation spanning the distance between the points.

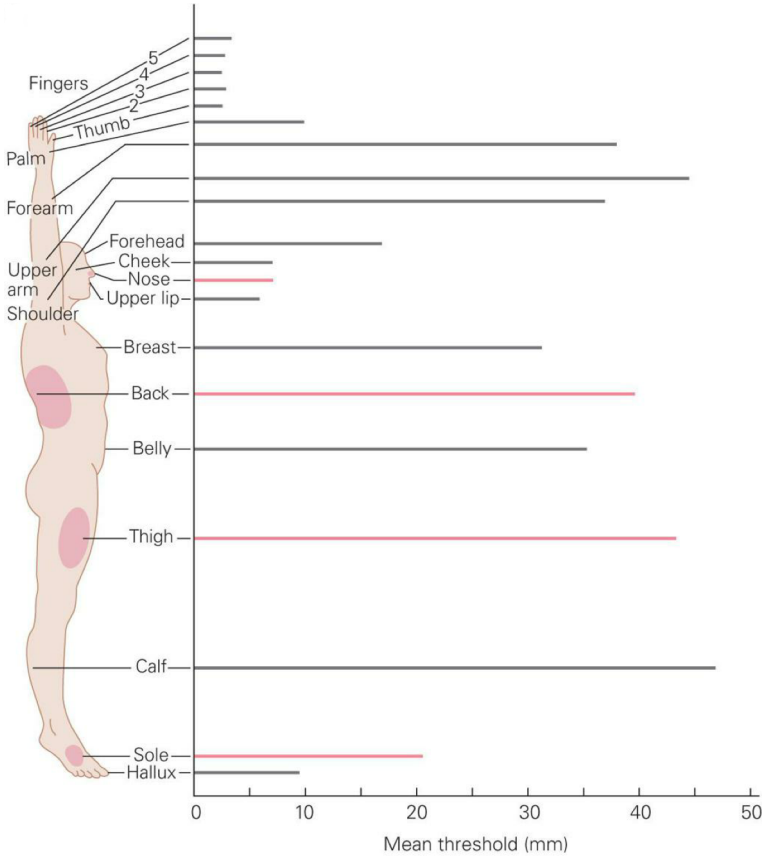


Figure 3.5: Two-point discrimination threshold [Kandel et al., 2013]

The belly has a good surface to sensitivity threshold ratio, the area is large enough to accommodate a substantial amount of actuators and offers a one-to-one mapping with the images obtained from the sensor. For optimum performance and taking into consideration the limitations of the human body, the actuators should be then placed approximately 35 mm apart from each other and their mechanical vibration should lie below 250 Hz.

Vibration Motors

Excentric Rotating Mass (ERM)

ERM vibration motors are used in most smartphones to provide vibration alerting

ERM vibration motors are the simplest and most popular kind of vibration motor available on the market. They are standard DC motors with an offset mass attached to the shaft that creates a centripetal force responsible for the vibration.

The behavior of an ERM vibration motor can be defined by the following equations:

$$f = \frac{\omega}{2\pi}$$

$$F = m \times a = m \times r \times \omega^2$$

$$F \propto V$$

f = vibration frequency (Hz), F = centripetal force (N), m = mass (kg), r = eccentricity of the mass (m), ω = angular velocity ($rad \cdot s^{-1}$), V = voltage.

For ERM vibration motors, the frequency and amplitude cannot be varied independently

The simplest way to modify the vibration amplitude is to modify the voltage applied to the motor, enabling the motor to deliver a varying amount of power which varies the angular velocity and therefore the centripetal force, varying the frequency of vibration proportionally. For ERM vibration motors, the frequency and amplitude cannot be varied independently.

The vibration amplitude, actually a measure of the acceleration, is commonly expressed in g ($1\ g = 9.8\ m \cdot s^{-2}$)

In order to regulate the voltage supplied to the motor, a Pulse Width Modulated (PWM) signal should be used. PWM is a type of modulation technique that, by altering a fixed DC voltage between bursts of "On" and "Off" at a fixed frequency, produces a different DC voltage that is proportional to the duty cycle. Since the frequency of vibration of a vibration motor is directly proportional to the voltage applied to it, PWM can be used to precisely control the motor.

There exist two different ERM vibration motor formats:

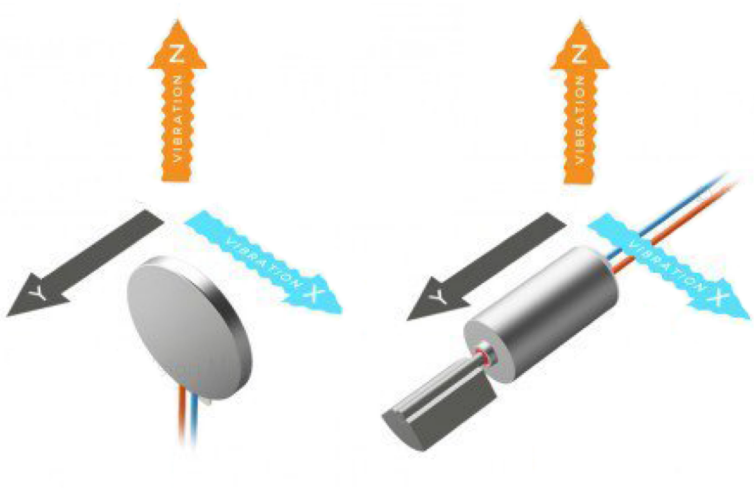


Figure 3.6: ERM axes of vibration [Precision Microdrives]

ERM vibration motors vibrate in two axes. Contrary to what it might appear, coin cell type ERM vibration motors are not a good candidate for this project because, if laid against the belly on the XZ axis, the vibration force would not be transmitted properly since it is only transmitted along the X and Z axes. On the other hand, a regular ERM vibration motor placed on the belly on the XY axis would work, since the vibration would be transmitted to the belly along the Z axis.

Average ERM vibration motors operate below the 250 Hz limit set by the human skin

Datasheets of regular ERM vibration motors show that at their maximum rated voltage, they operate at around 11,000 to 14,000 rpm or, in other words, 183 to 233 Hz. This vibration frequency lies below the 250 Hz limit set by the sensitivity of the human skin, making them a good candidate for this kind of application.

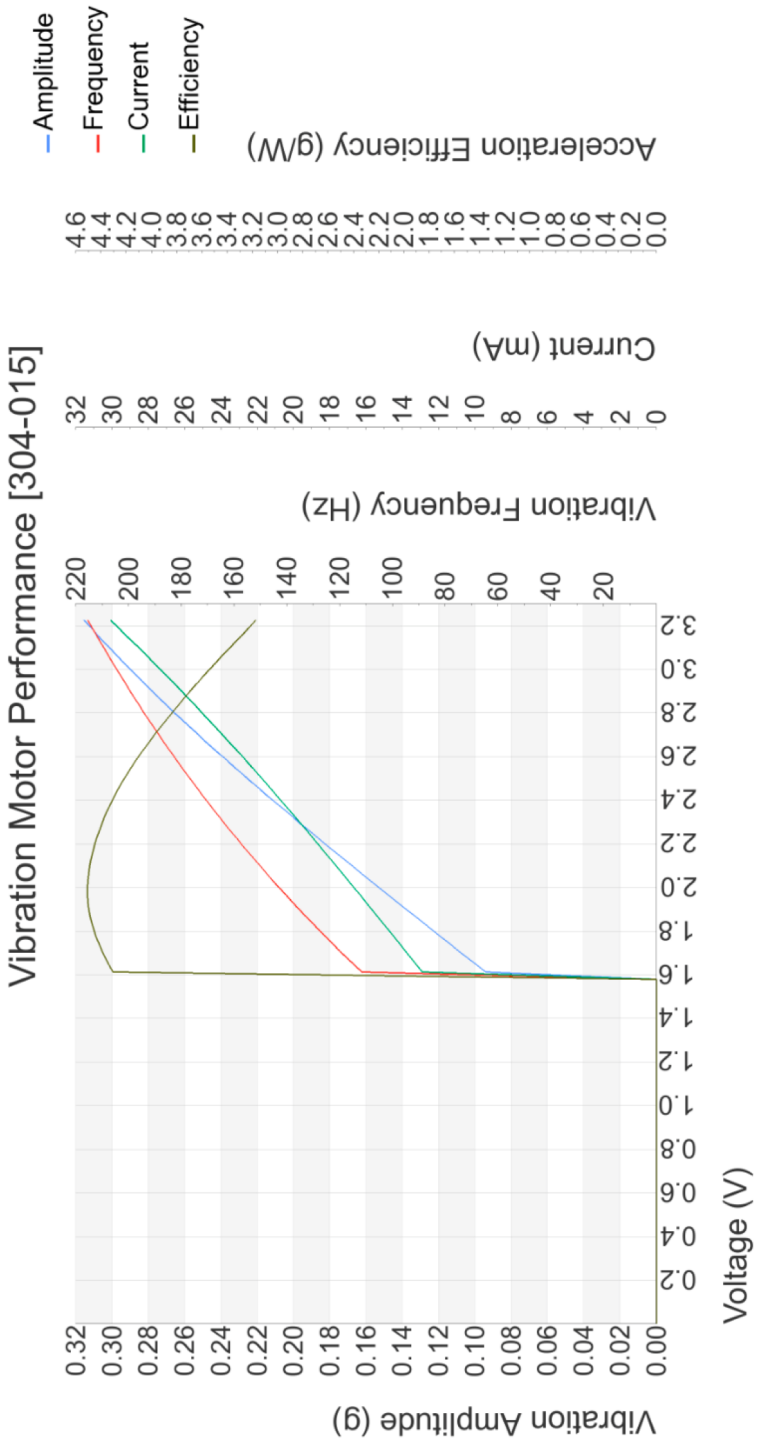


Figure 3.7: Performance of an average ERM vibration motor [Precision Microdrives]

Linear Resonant Actuator (LRA)

The working principle of LRAs is the similar to that of a speaker

LRAs are an alternative to ERM vibration motors. While ERMs rotate an unbalanced mass to create a centripetal force, LRAs have no external moving parts. Their working principle based on an internal magnetic mass attached to a spring. An electrical signal through the LRAs coils forces the mass up and down, resulting in a force that causes displacement. The mass and spring system is a basic example of a Simple Harmonic Oscillator, which displays resonance. This means that the device oscillates with a greater amplitude at particular frequencies. As the magnetic mass is driven in each direction the LRA requires an AC signal, requiring an dedicated integrated circuit (IC) to drive them and thus making it more difficult to drive than ERM vibration motors.

Generating AC signals is not a trivial task

However, LRAs are designed to be as efficient as possible. Both the bandwidth and the energy loss are described by the Q or quality factor. For the case of LRAs, their high Q factors indicate a low rate of energy loss but a very narrow bandwidth. Datasheets of regular LRAs show resonant frequencies ranging from 175 to 235 Hz, again below the 250 Hz limit.

LRAs vibrate only in one axis. For the case of OpenVNAVI, the best option would be using LRAs that vibrate on the Y-Axis.

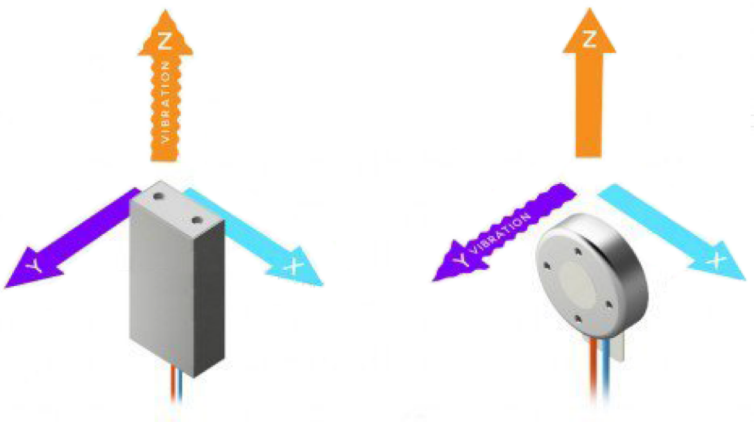


Figure 3.8: LRA axes of vibration [Precision Microdrives]

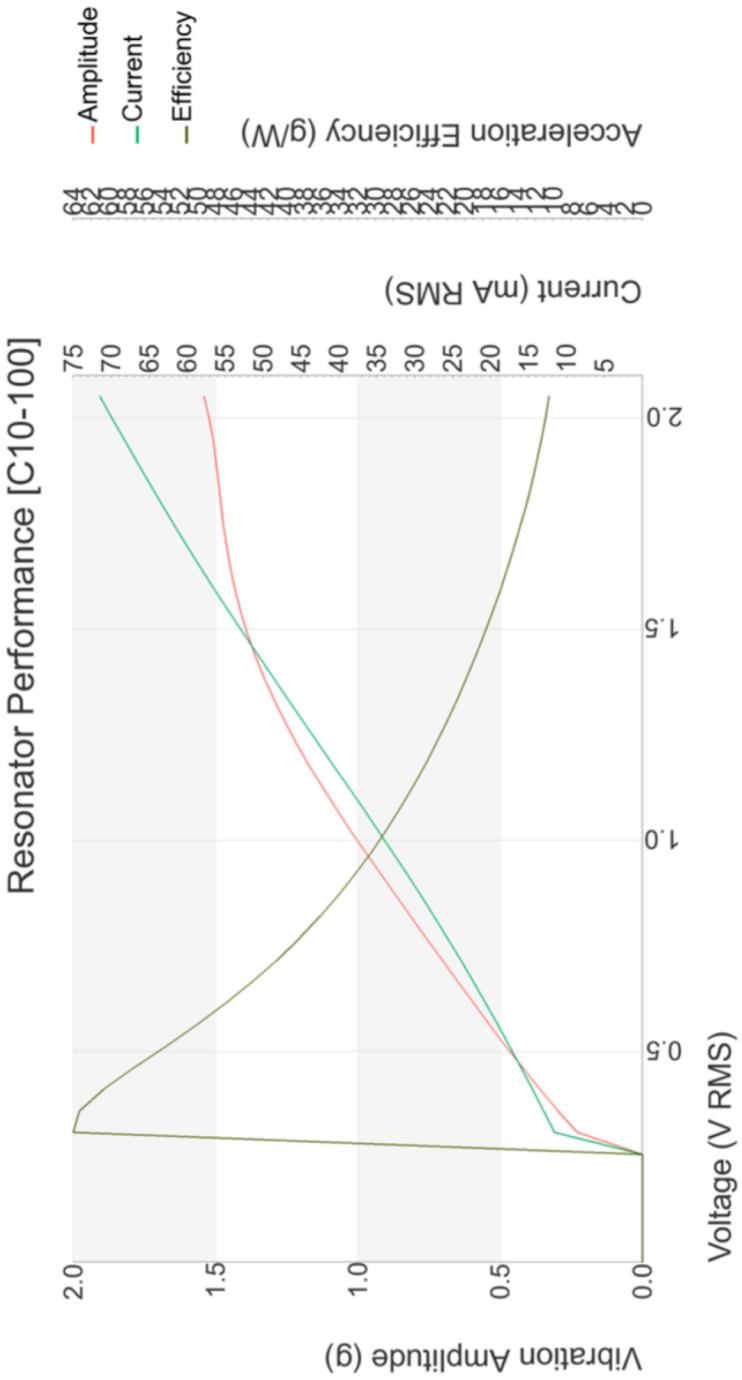


Figure 3.9: Performance of an average LRA [Precision Microdrives]

LRAs have much longer lifespans, they are more compact, more durable and have no external moving parts. Unfortunately, they are much less common and require a dedicated IC per motor, thus making them a much more expensive alternative.

Taking a sample of different ERM vibration motors and LRAs available on the market, a comparison of both technologies is performed:

	ERM	LRA
Vibration amplitude range (g)	0.6 - 6	1.4 - 1.5
Vibration frequency range (Hz)	183 to 233	175 - 235
Rated voltage (V)	1.4 - 3.4	0.2 - 2
DC PWM voltage	Yes	No
Dedicated driver	Optional	Yes
Cost	Low	High

Table 3.4: Comparison vibration motor technologies

Both ERM vibration motors and LRAs are valid technologies for this kind of application. However, cost is an important factor to consider when making the final decision. The need for a dedicated IC for driving each LRA would increase the cost dramatically, therefore the best option in this case would be to use ERM vibration motors.

Other technologies

Other technologies considered for this project include piezo elements driven with PWM at a low frequency. No tests were performed due to the fact that the piezo elements had to be in contact with the skin for optimum transfer of the vibration force, reducing their performance.

Using non-established technologies would require thorough testing

Local electrostimulation was also considered. In this case potential issues like calibration, the level of invasiveness of the electrodes and the discomfort caused by this setup discouraged the use of this technology.

Jones et al. [2004] developed an experimental tactor array using NiTi Shape Memory Alloys (SMA) but concluded that the tactile sensations produced by the SMAs were not well localized.

Drivers

I2C is a serial bus used for attaching lower-speed peripheral ICs to processors and microcontrollers

Driving the ERM vibration motors with a PWM voltage is the most convenient way to do it but the number of available PWM outputs on the Raspberry Pi 2 is limited. Fortunately there exist dedicated drivers and other ICs that, by communicating with the Raspberry Pi 2 via the I2C protocol, would enable the Raspberry Pi 2 to control a higher amount of ERM vibration motors.

Texas Instruments DRV2605
The DRV2605 is a haptic driver designed for ERM vibration motors and LRAs. It controls ERM vibration motors via PWM and supports the I2C communication protocol. It is designed to deliver high-end haptic response thanks to its waveform library, active braking and overdrive modes, enabling a precise control of the haptic response. The cost of each DRV2605 IC is higher than other available alternatives because it includes a fully-paid license to Immersion Corp, a company that owns most haptics-related intellectual property.

Each DRV2605 IC would only be able to control one ERM vibration motor. This fact, along with its higher price point and the fact that the waveform library, active braking and overdrive modes are not essential for the current implementation of OpenVNAVI, show that this IC might not be the best choice for this project.

NXP PCA9685
The PCA9685 is a I2C-controlled 16-channel PWM LED driver with 12-bit duty cycle control. Although this driver is designed for driving LEDs, it can be used to drive ERM vibration motors with the help of discrete electronics such as N-channel MOSFETs. The fact that this IC is able to potentially drive 16 ERM vibration motors and its low price point makes it a very good candidate for OpenVNAVI. One disadvantage is the need to send the PWM signal individually to each vibration motor, increasing the number of cables in the system.

The company Adafruit sells a servo driver based on the PCA9685 and offer a Python library under a BSD license to control it. This other factor makes it a very good candidate for OpenVNAVI.

Adafruit is an open-source hardware company that designs and manufactures electronics and provides learning resources

Texas Instruments TLC5940

The TLC5940 is a SPI-controlled 16-channel PWM LED driver with 12-bit duty cycle control. Similar to the PCA9685 but using the SPI communication protocol instead of I2C, the TLC5940's only disadvantage compared to the PCA9685 is a slightly higher price point and the lack of a Python library to control it.

Worldsemi WS2811

This IC is currently used by the company Adafruit to control their NeoPixel LEDs. The fact that it only needs one PWM GPIO pin of the Raspberry Pi 2 and only a 1-wire data bus, along with the availability of Python libraries and great community support makes it a very good candidate for OpenVNAVI.

GPIO stands for general-purpose input/output

The WS2811 is able to control one RGB LED, thus, offering the possibility of controlling from one to three ERM vibration motor units per IC with the help of N-channel MOSFETs. The implementation of these drivers would require one IC per vibration motor unit, increasing the total cost of the device, but dramatically reducing the amount of cables needed for transmitting the PWM signal compared to the PCA9685 since the WS2811, alongside the DRV2605 could be integrated into each vibration motor unit, but the PCA9685 drivers would need to be placed on a different PCB, far away from the vibration motors.

Using one WS2811 per motor would reduce the number of cables required

The following table summarizes the characteristics of the most relevant drivers presented above:

	DRV2605	PCA9685	WS2811
Bus cables reaching the array	2	16 x No. ICs	1
Channels	1	16	1
Price (per ch.) (€)	2.50	0.10	0.50

Table 3.5: Comparison of drivers

Both the PCA9685 and the WS2811 are good candidates for driving the motors

Analyzing the characteristics of all these different options it can be concluded that the best candidate for this project is the NXP PCA9685 driver, being the Worldsemi WS2811 another good option to consider. Both have Python libraries available and would also deliver the same results. Although the PCA9685 would require a large amount of cables to drive all vibration motors, the cost would be lower than using WS2811 drivers, making the PCA9685 the best option for a first prototype.

Optimizing for the value of the two-point discrimination threshold on the belly of 35 mm and its average surface area, the author concluded that the number of ERM vibration motor units to use is 128, or a 16x8 array, requiring 8 NXP PCA9685 driver ICs.

The use of decoupling capacitors is essential for maintaining a clean voltage signal

Each vibration motor should be soldered on a PCB with a decoupling capacitor and an N-channel MOSFET driven by the PWM signal. This PCB should be connected to 5 V, ground and the PWM signal coming from the PWM driver. It should be housed in a 3D-printed housing to isolate the moving parts of the ERM vibration motor from the outside and provide firm contact against the belly. The 8 PCA9685 drivers should be soldered on a custom PCB according to the manufacturer's specifications and connected to the Raspberry Pi 2 via I2C.

There exist many different ERM vibration motor manufacturers on the market offering vibration motors of varying characteristics. The process of selecting the right vibration motor for this project would require testing individual samples. This process is described in the implementation.

3.3.4 Power Source

The technical specifications of the vibration motors selected for this project show a current consumption of 100 mA at their maximum operating voltage of 3 V. Knowing the current consumption of the rest of the components the total current consumption is obtained as follows:

Components:

- Raspberry Pi 2: 200 mA
- ASUS Xtion PRO LIVE: 500 mA
- Vibration Motor Unit: 100mA

Maximum current consumption:

$$200 + 500 + (128 \cdot 100) = 13.5A$$

These result corresponds to the theoretical worst case scenario in which all vibration motors are working at their maximum operating voltage. Although current consumption on the device may probably be lower, the electronics should be dimensioned to allow for such current demand.

The theoretical maximum current demand is very high for a mobile device

This instant current demand is substantial for a mobile device and cannot be supplied with regular batteries. LiPo RC batteries are able to provide enough discharge rate to meet those demands and could be easily implemented but for the case of this prototype and to save costs, power will be supplied using a 12 V bench power supply.

The issue of the current consumption should be addressed for future implementations. A drastic reduction in current consumption is vital to make the device fully mobile.

Buck converters are switching regulators with an efficiency above 95%

The Raspberry Pi 2 and the PCA9685 PWM LED drivers operate with an input voltage of 5 V, for this reason a DC/DC buck converter is necessary to regulate the voltage down to 5 V for both powering the device with batteries and with a power supply. Any off-the-shelf buck converter with an input voltage range of 5 - 15 V and a current rating greater than 13.5 A would work. In this case an affordable DealExtreme Jtron converter with the following characteristics was selected:

- Input voltage range: 4 - 32 V
- Output voltage range: 1.2 - 32 V
- Maximum operating current: 15A

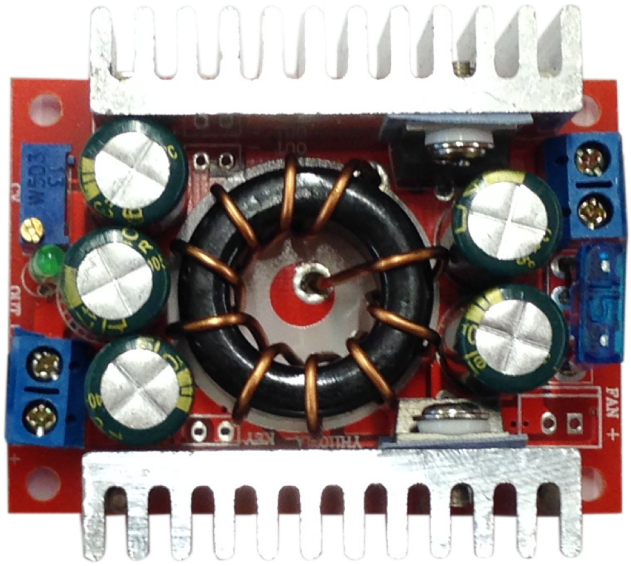


Figure 3.10: Jtron buck converter

3.3.5 Driver Unit

A custom PCB is required for accommodating the PCA9685 PWM LED driver ICs, output pins, screw terminals and conditioning circuits for the input and outputs of the Raspberry Pi 2. The Raspberry Pi 2 GPIOs use 3.3 V logic levels and they can be damaged if connected directly to 5 V levels. For this reason the PCB will need to accommodate a voltage divider for each of the 5 GPIO inputs.

The Raspberry Pi 2 can also be powered in different ways. The author chose to power it using Micro-B USB connector for safety reasons given that the Raspberry Pi has polyfuse protection on that input port.

The safest way to power the Raspberry Pi 2 is via the USB port

The list of elements required in the enclosure that need to be implemented is listed below:

- 5 general purpose push buttons
- 3 general purpose LEDs
- 1 power switch
- 1 battery/power supply selector switch
- 1 speaker output

The connections from this PCB to Raspberry Pi 2 that need to be implemented are listed below:

- Raspberry Pi 2 power
- I2C bus
- 5 inputs for Normally Open push buttons
- 3 outputs for LEDs
- 1 speaker output

This PCB will also need to have the necessary screw terminal connections to properly handle the connections between the buck converter and the power buttons, as well as reverse polarity protection with a P-channel MOSFET.

Without protection, the circuits could be damaged if the input polarity is reversed

3.3.6 Enclosure

3D printing is a cheap and effective method for prototyping

The enclosure should contain the electronics, input and output devices, switches and all the cables that connect them to Raspberry Pi 2, the Driver Unit and the actuators, leaving a space for them to pass through. Another outlet must be left for placing a male-to-female Ethernet cable for easier access to that port from the outside of the enclosure. 3D-printing using PLA plastic would be the best method for manufacturing it due to its reduced complexity and cost.

3.3.7 Vest

The vest hast to be custom built for this prototype

The structural support for all the components of OpenVNAVI should offer enough structural integrity to hold them in position in both static and dynamic conditions. It should be made for users with different chest, waist and hip measurement sizes and it should also be comfortable to wear. For an optimum performance the vest should provide a tight fit so that the vibration motor units, attached to the vest with hook and loop fastener, colloquially known as velcro, transmit all the vibration force to the user. For this reason, semi-stretch fabric is needed. Due to the particular requirements of the vest, the optimum solution was to build one using semi-stretch, 2% elastane 98% cotton fabric, thread and velcro.

Elastane fibers provide flexibility to the fabric

3.3.8 Code

As previously discussed, due to time constraints for the completion of this Bachelor Thesis, the main purpose of this system would be obstacle avoidance. To achieve this, the main loop of the Python program must capture a frame from the depth sensor, downsample the image from 640x480 to 16x8 pixels and store it in an array. After that, a mapping function must transform the 8-bit depth information into 12-bit PWM values within the range of the vibration motors and then apply that value to each one of them.

3.4 Implementation

3.4.1 Depth Sensor

For optimum scanning of the environment and in order to integrate it as much as possible with the vest, the sensor was positioned onto the user's chest. The base plate of the sensor was removed and the device was placed on 3D-printed support attached to the vest.

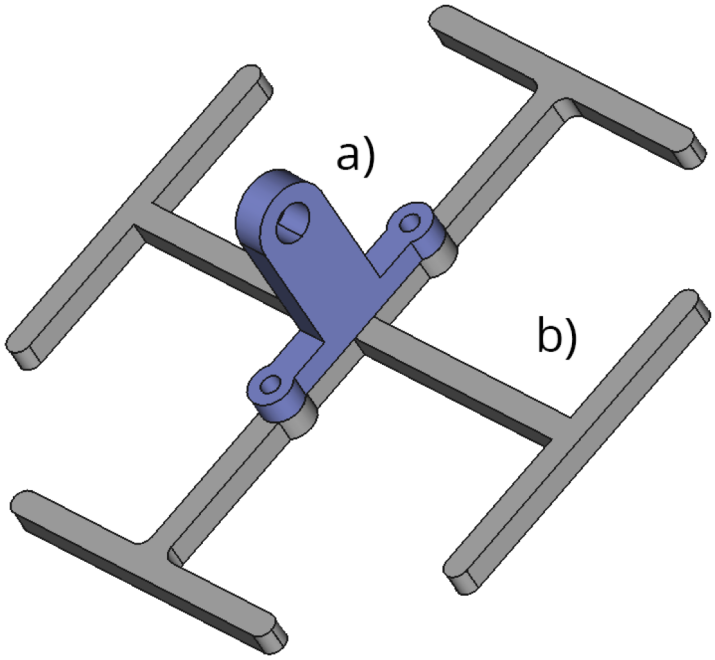


Figure 3.11: Support for the sensor

Part **a)** of the support allows for vertical pivoting of the sensor and part **b)** adds stability by helping to restrict two more degrees of freedom when the vest is pressed against the chest.

3.4.2 SBC

For the installation of the operating system on the Raspberry Pi 2 a SanDisk Ultra 16GB Class 10 microSDHC card was chosen. The specifications for class and storage space lie above the minimum requirements specified by the Raspberry Pi Foundation. Raspbian was the operating system of choice due to its popularity. A clean install was performed and GPIO and I2C were configured.

Python is a popular programming language in the Raspberry Pi community

There exist different ways to interface with the ASUS Xtion PRO LIVE but not all of them worked on the Raspberry Pi's ARM architecture. Python was chosen as the programming language for this project because it is a popular programming language in the Raspberry Pi community, but this decision limited the amount of possible options to interface with the sensor.

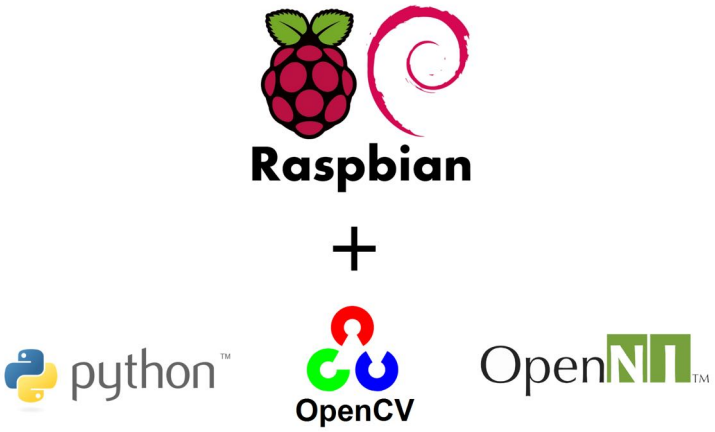


Figure 3.12: Open-source software

OpenNI 1.x + PyOpenNI
OpenNI is an open-source framework developed by PrimeSense and currently maintained by Occipital that provides a set of APIs with support for voice and command recognition, hand gestures and motion tracking. There exist no official Python bindings for this version of OpenNI but there is an open source project called PyOpenNI that serves that purpose.

OpenNI 1.x has been unmaintained since 2013

This combination works fine when compiling on a x86 or x64 architecture, but it was not possible to compile PyOpenNI for the Raspberry Pi's ARM architecture. OpenNI 1.x stopped being supported in 2013, hence the lack of maintenance of PyOpenNI.

OpenNI 2.x + PrimeSense Python bindings
As in the previous case, this configuration works on both x86 and x64 architectures, but there was a problem when trying to install the official PrimeSense Python bindings on the Raspberry Pi. OpenNI 2.x and the official Python bindings stopped being maintained by PrimeSense when the company was acquired by Apple in 2013.

OpenNI 2.x continues to be used by various manufacturers today

OpenNI 1.x + OpenCV
OpenCV offers the possibility of capturing data via OpenNI 1.x. This combination works with the Python version of OpenCV 2.4.11 but only by previously defining some constants in the program so that the VideoCapture function of OpenCV can access the sensor. These constants are defined in the C++ version of OpenCV 2.4.11 but not in the Python version and they are crucial for it to work [Howse, 2013].

OpenCV needs to be configured to be able to run with OpenNI

First OpenNI 1.5.7.10 was compiled from source with some small modification to the code to allow it to compile on the Raspberry Pi 2's ARM architecture. After installing OpenNI 1.5.7.10, OpenCV 2.4.11 was compiled from source enabling the option "`WITH_OPENNI=YES`"

3.4.3 Actuators

Aliexpress offers affordable products manufactured in China

As previously discussed, there exist many different ERM vibration motor manufacturers on the market offering vibration motors of different characteristics and prices. For the process of selecting the right vibration motor, a non-exhaustive test was performed on different ERM vibration motors sourced from AliExpress.

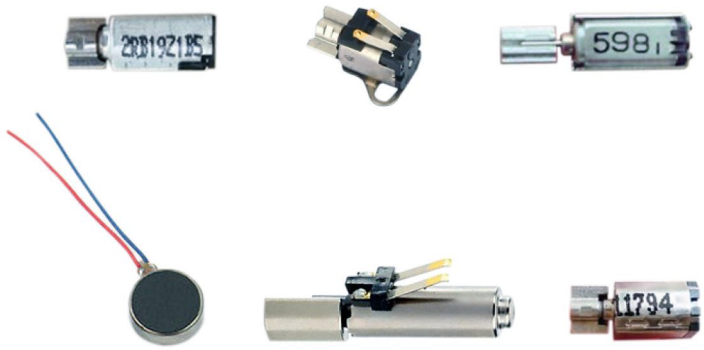


Figure 3.13: ERM vibration motor samples

The performance of the different ERM vibration motors was very dissimilar

The motors were tested for operating voltage and current range, apparent vibration amplitude at their maximum rated voltage and performance under stress conditions. The ERM vibration motor with the best performance was the one with serial number 2RB19Z1B5, showing a voltage range of 0.5 to 3.3 V, a current range of 18 to 100 mA and excellent performance compared to the rest.

Due to the nature of the majority of products sourced from Chinese distributors on AliExpress it is hard to find products with brand identity and datasheets, consequently this is the only information available to identify this particular ERM vibration motor. On the other hand, they are substantially cheaper compared to other known brands.

The dynamic behavior of a linear, time-invariant system such as a DC motor driven by a PWM signal will exhibit characteristics such as noise and inductive voltage spikes. These issues need to be addressed in order to obtain a clean signal throughout the system and avoid issues when all 128 motors are functioning at the same time.

The negative voltage spikes are caused by the inductive load of the motor when its supply voltage is suddenly removed

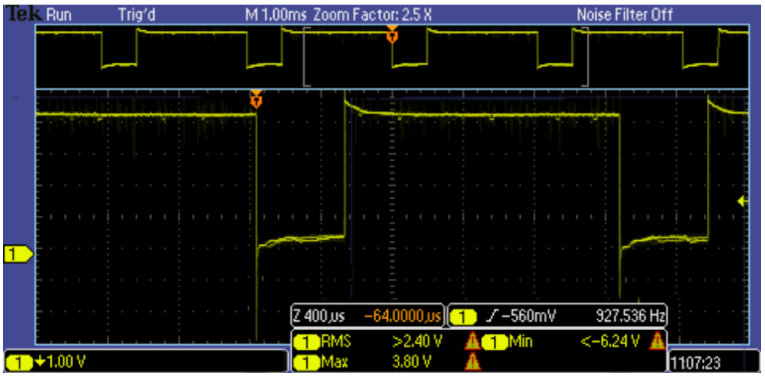


Figure 3.14: Vibration motor with no filtering

The PCA9685 driver is able to offer 25 mA of output current per channel, which is not enough to drive the vibration motor at its operating voltage. For this reason, the motor needs to be driven with a transistor. The most popular method for driving small vibration motors is to use an MOSFET, which generally have lower R_{DS} and V_{GS} enough to easily drive the MOSFET in saturated mode of operation. An N-channel MOSFET is used because the drive signal from the PCA9685 is active-high. BJT transistors can also be used but they are less efficient and require heat-sinking. A general-purpose 2N7002 N-channel MOSFET was selected.

An N-channel MOSFET is used because the drive signal from the PCA9685 is active-high

A capacitor is needed to reduce the electromagnetic interferences (EMI), high frequency electromagnetic wideband noise generated by the motor. A 100 nF capacitor exhibited the best results after empirical measurements with different capacitance values. This value is small enough not to interfere with a low/medium frequency switching signal but large enough to limit the voltage spikes from the motor caused by sparks in the commutator.

A flyback diode is necessary to protect the MOSFET against voltage spikes from the coils. As with the case of the MOSFET, a generic 1N4148 high-speed diode was selected.

A pull down resistor is also required to reduce quiescent current and keep the MOSFET fully off when the active signal is not present. Since MOSFETs are charge controlled devices, without this resistor a gradual charge can build up on the gate and cause sporadic operation of the motor. A standard 47 kΩ resistor was selected.

After deciding which components to used, the final schematic of the motor unit was designed and tested.

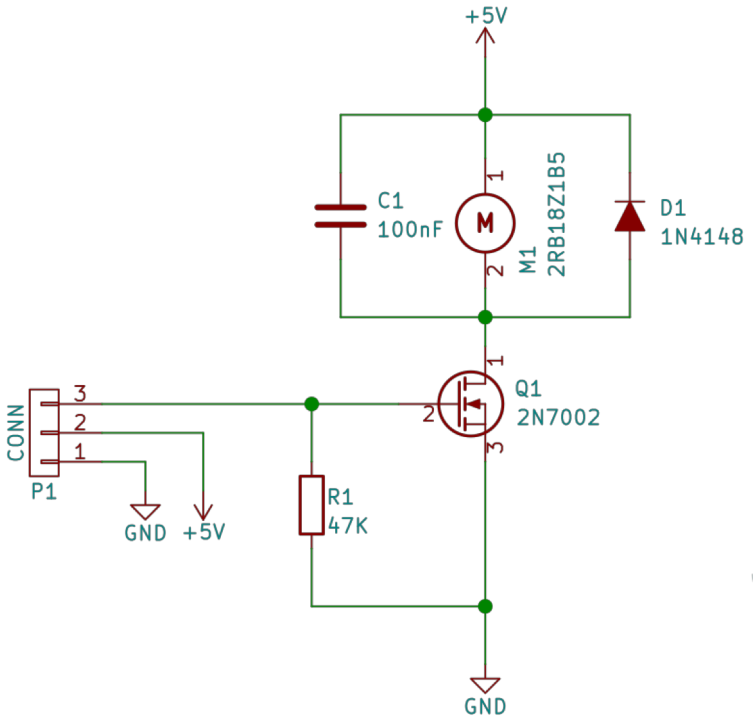


Figure 3.15: Motor unit schematic

With all the these components installed, the signal appears to be much cleaner than with the ERM vibration motor alone:

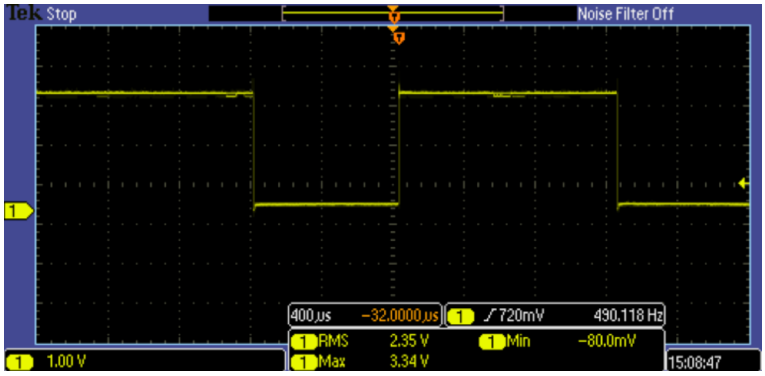


Figure 3.16: Vibration motor with filtering

After analyzing the performance of the components of the motor unit, the PCB was designed using KiCad.

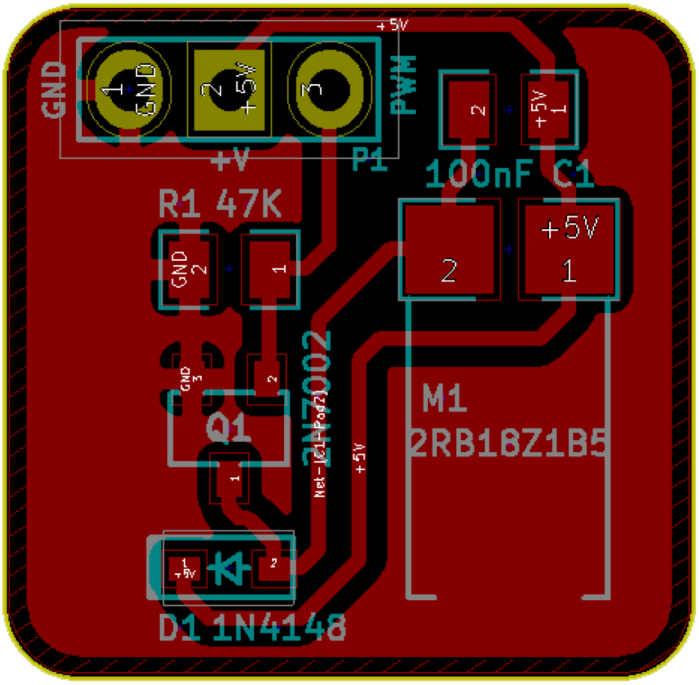


Figure 3.17: Motor unit PCB

Smaller and more robust connectors like JST or Molex PicoBlade are much more expensive than regular pin headers

The PCBs were milled in the Fab Lab using the PCB milling machine and the components were soldered manually. To allow for easy way to replace the motor unit PCBs in case of a malfunction, a connector was installed. The author chose a regular 3-way male angled pin header for the PCB for reasons of cost.

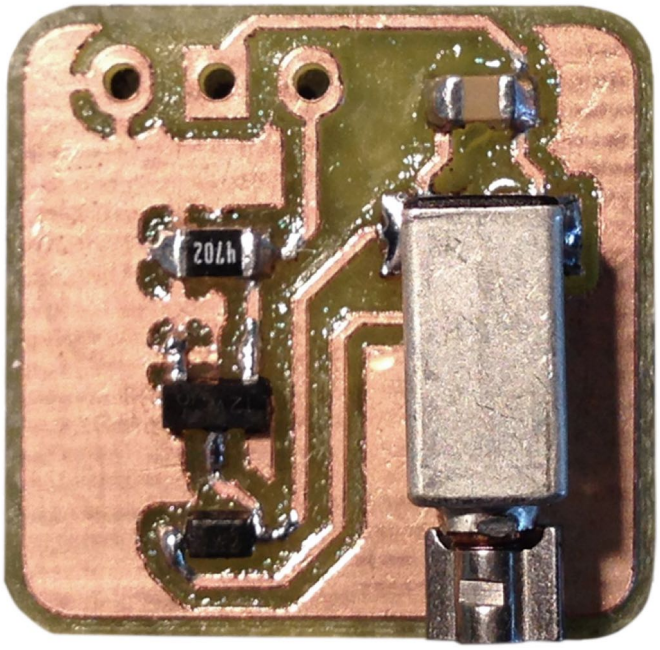


Figure 3.18: Assembled PCB

The 3D-printed housing of the motor unit was designed for enclosing the moving parts and for allowing a firm and efficient transfer of the force to the skin. A small cover secures the PCB in position and a small slot allows for a pair of pliers to remove the PCB in case of failure.

By reducing the mass of the motor unit, the vibration amplitude would increase

Using newton's second law of motion, $F = m \cdot a$, knowing that at a fixed point in time the vibration motor creates a steady centripetal force and that the vibration amplitude is a measure of acceleration; in order to increase the acceleration of the motor unit, the mass of the 3D-printed housing should be as small as possible.

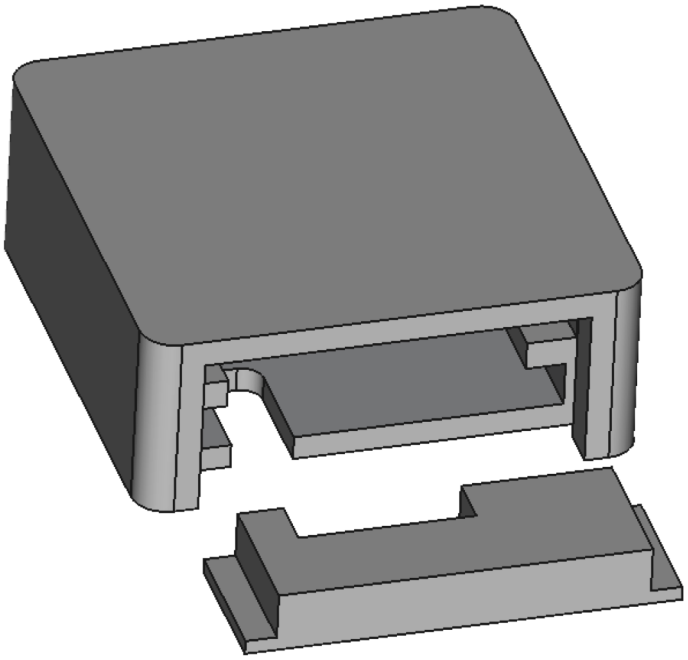


Figure 3.19: Motor unit enclosure 3D design

Figure 3.20: Motor unit enclosure

The motor units are attached to the vest using velcro. An adherence test between PLA and the velcro was performed to check the behavior of different adhesives. The best results were obtained using Pattex 100% multi-power.

Figure 3.21: Velcro-PLA adherence test

Figure 3.22: Motor unit enclosure with velcro

The motor units need to be calibrated before assembling them

The tolerances of the 3D printer as well as the fact that the enclosures were printed without support material, created a calibration issue. The vibration amplitude varies as a function of how tightly secured the vibration motor is to the enclosure. If the vibration motor is too loose, it would rattle, creating a disturbing sound as well as losing efficiency in the transfer of the force.

The simplest, most effective way of solving this issue was to use small pieces of 80 $g \cdot m^{-2}$ paper placed underneath the ERM vibration motor, raising it in 0.1 mm increments until it was tightly secured.

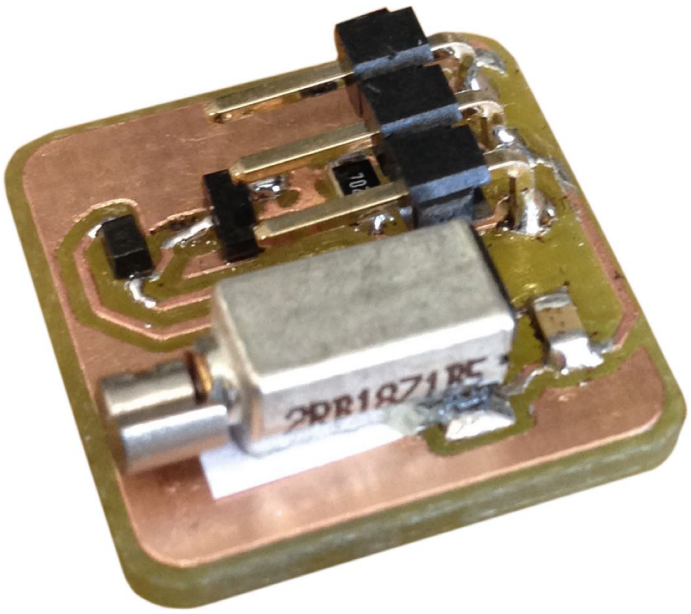


Figure 3.23: Calibration wedge

3.4.4 Power Source

The buck converter did not require any extra work other than calibrating the output to deliver 5 V and performing the appropriate connections to the Driver Unit.

3.4.5 Driver Unit

Overclocking the Pi would require a fan for cooling the CPU

The Driver Unit was designed according to the established requirements. The PCB holes were designed to match the Raspberry Pi 2's so that the Driver Unit PCB could be mounted on top of it. Two pins for powering a fan were placed in case the Raspberry Pi 2 needs to be overclocked in the future.

The inputs were equipped with a voltage divider to adjust the voltage from 5 V to 3.3 V to make it compatible with the Raspberry Pi 2's GPIO.

$$V_{OUT} = \frac{R_2}{R_1 + R_2} \cdot V_{IN} = \frac{10000}{5100 + 10000} \cdot 5.0V = 3.3V$$

The 5K1Ω resistors would also limit the current, protecting the GPIO pins. Taking into account the resistor tolerances and using $\pm 1\%$ resistors, the output voltage would always be within the specifications.

The outputs for the LEDs were equipped with 220Ω resistors for convenience, two 10KΩ pull-up resistors were also placed along the I2C bus according to the I2C specifications [NXP] and a $10\mu F$ decoupling capacitor, as well as a 10KΩ connecting pin 23 and GND for each PCA9685 IC were placed according to the manufacturer's specifications.

The schematics of the Driver Unit can be found in Appendix C "Driver Unit Schematics".

The 2-layer PCB was designed and milled following the same procedure as with the Motor Unit PCBs.

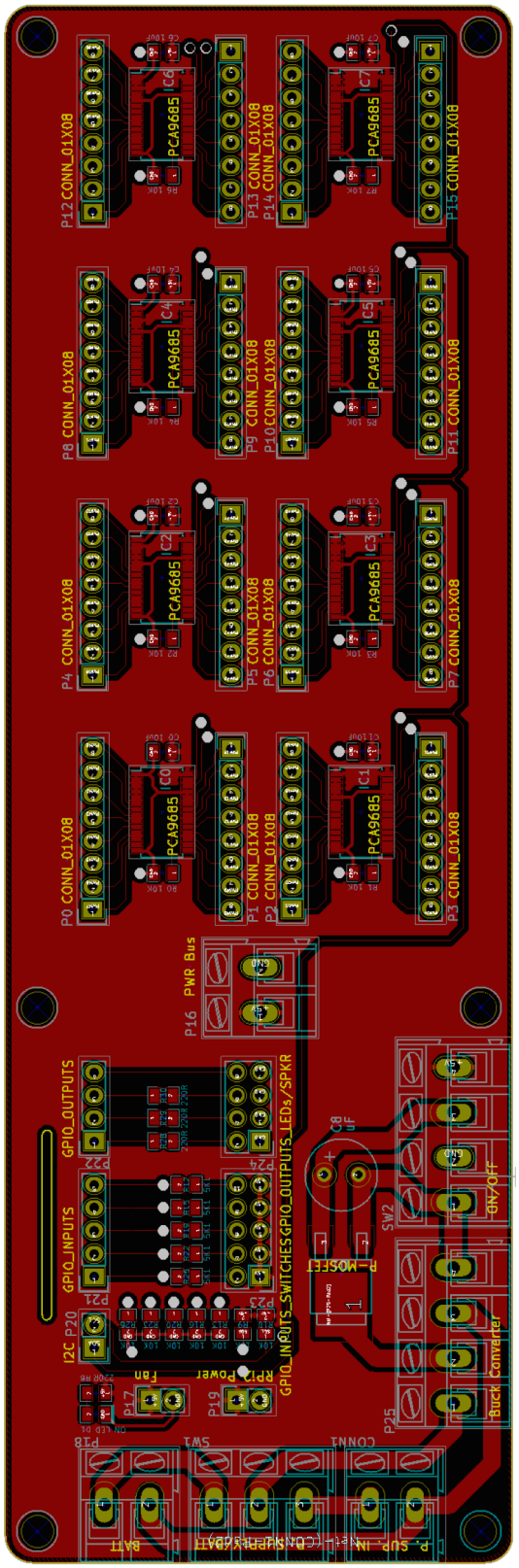


Figure 3.24: Driver Unit PCB

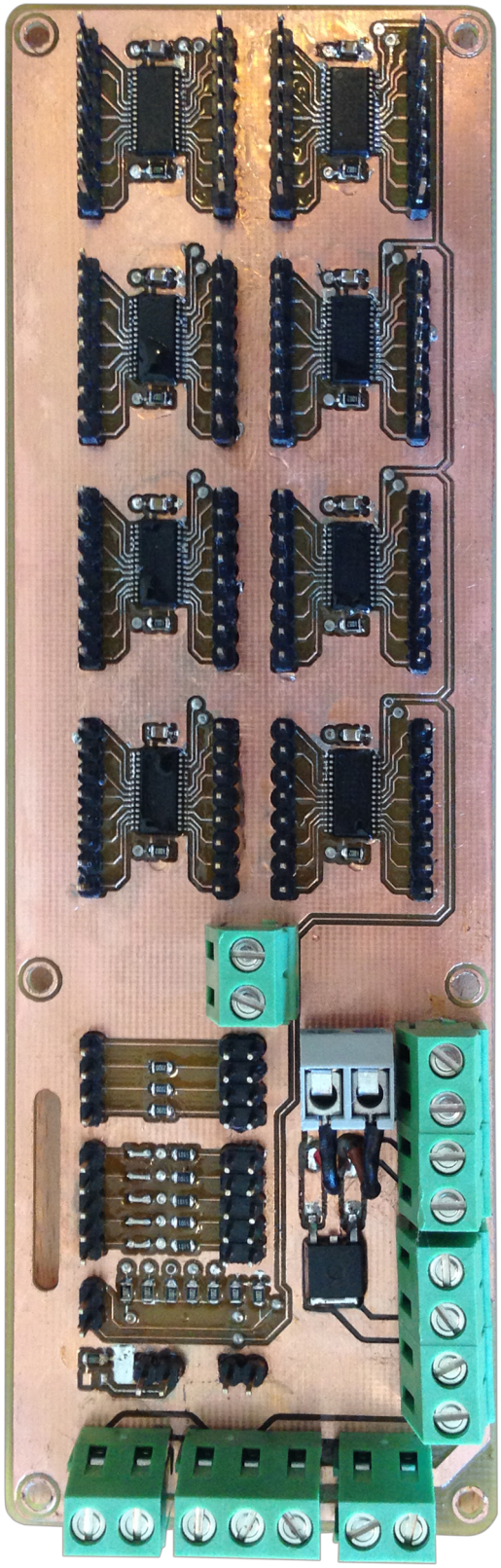


Figure 3.25: Driver Unit PCB

3.4.6 Enclosure

The 3D-printed enclosure was designed according to the specifications. The dimensions of the outer holes are based on the buttons and switches used for this project but they may probably be different for other implementations if different manufacturers and models are used. The enclosure has four screws for securing the Raspberry Pi 2 and the Driver Unit, as well as another four screws for the cover.

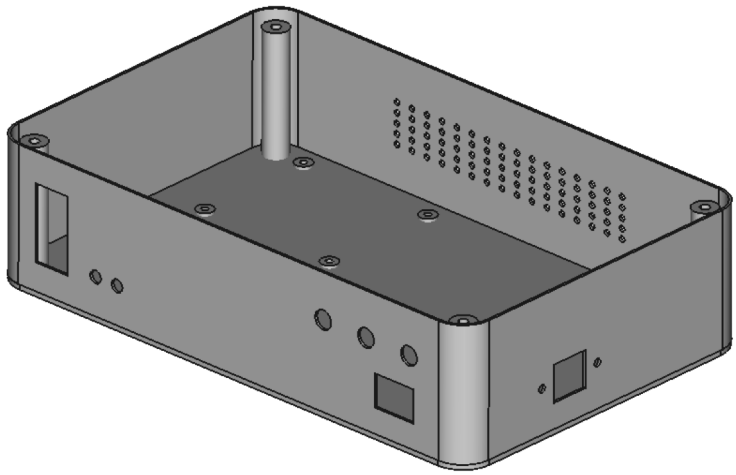


Figure 3.26: Enclosure

3.4.7 Vest

The aforementioned materials needed for the vest can be found in any local fabric supplier. The elements purchased were a piece of 1.3x1.5 m dark gray 2% elastane 98% cotton fabric, dark gray thread and 9 m of velcro. 16 velcro strips of 36 cm in length spaced 3 cm from each other were sewn to the vest.

Figure 3.27: Fabric, thread and velcro

Fabric strips were placed in-between the velcro strips to serve as cable guides. Other cable guides were also placed on the interior of the vest to guide the cables over the shoulders. Horizontal velcro strips on the exterior allow the vest to adapt to different chest, waist and hip measurement sizes.

The complete step-by-step process can be found in Appendix D "Making the Vest".

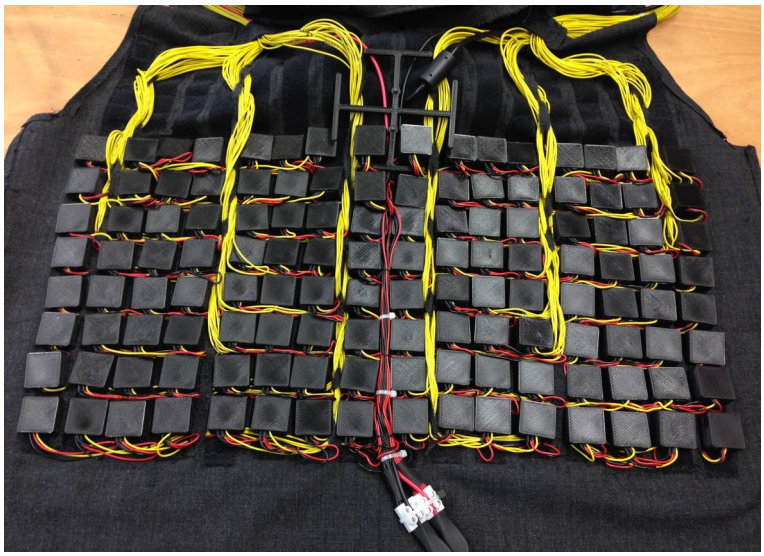


Figure 3.28: Vibration motor array

Figure 3.29: Vest (front)

3.4.8 Code

The code was written following the specifications, using the subsequent structure:

- Import of Python modules.
- Definition of constants.
- Definition of functions.
- Initialization.
- Main loop.

The main loop initializes the PWM drivers, the depth sensor and the GPIO and waits for a start/stop switch to be pressed. After that it calls the function `setVibration()` that captures the depth information and sets the appropriate PWM values to each motor, handling the interrupts accordingly.

The test functions were used for calibration and performance measurement purposes

The test function `depthTest()` grabs frames from different depth information channels, multiplies the output by different constant gain values and saves the frames for visual inspection of the range of depth values of a scene. The channel and gain values that provided the best contrast were selected. Other test functions such as `sweepTest()` and `strobeTest(row)` test the performance of the vibration motors.

Different downsampling algorithms provide different results

The choice algorithm used for downsampling the depth frame from 640x480 to 16x8 pixels is not a trivial decision as it would influence the results and the performance of the device. Some algorithms aim for preserving hard edges while others provide results with smoother gradients.

The scaling algorithms available in OpenCV's `resize()` function are nearest-neighbor interpolation, bilinear interpolation (by default), pixel area relation, bicubic interpolation and Lanczos interpolation.

Original Image

Nearest-neighbor

Linear

Pixel Area Relation

Bicubic

Lanczos

Figure 3.30: Scaling Algorithms

After visually inspecting the results it can be concluded that bilinear interpolation, bicubic interpolation and Lanczos interpolation offered similar quality results, preserving the edges and the definition of the objects of interest better than the other methods. The default scaling algorithm, bilinear interpolation, was used for this project.

The code can be found in Appendix A "Code".

Chapter 4

Evaluation

In this chapter, after setting the requirements and implementing all the features, the performance of OpenVNAVI and each of its components is evaluated.

4.1 Performance

General performance tests conducted in a standard indoors environment showed that the obstacle detection function implemented in OpenVNAVI works according to the expectations, although several improvements should be carried out in order to fully meet the requirements.

Small obstacles close to the floor are difficult to detect

Common obstacles found indoors such as walls, doors, tables, chairs, etc. could be easily detected and avoided, showing promising results. Nevertheless, small obstacles close to the floor failed to be recognized due to the fact that when the high resolution image is downsampled to 16x8 pixels, that information is lost.

4.1.1 Depth Sensor

Known limitations of the ASUS Xtion PRO LIVE such as decreased performance outdoors, when measuring depth of objects with sharp angles with respect to the normal of the sensor and around transparent materials such as glass were confirmed. These known issues were not conducive to a decrease in performance but they might need to be addressed in further iterations to guarantee a solid performance in every possible situation a BVI user might encounter.

4.1.2 SBC

A faster SBC is needed for achieving better performance

The Raspberry Pi 2 performed very well in terms of compatibility, flexibility and power usage, but the performance of the system in terms of frames per second (fps) was not as good as expected, experiencing an average performance of 6 fps. As examined later in this chapter, the code was initially ruled out of the possible causes of this poor performance, leaving the performance of the CPU of the SBC as the principal cause.

4.1.3 Actuators

The ERM vibration motors provided excellent vibration amplitude, even through clothing, and the vibration motor unit enclosures adapted very well to the curvature of the belly area. Nearly 100% of the motors passed the stress tests, although some of them experienced random activations probably due to overheating and the inconsistencies of the manual soldering process.

The small calibration issues described in the previous chapter were only partly solved. Achieving a uniform calibration was not possible using discrete steps of 0.1 mm (for 80 $g \cdot m^{-2}$ paper). Overall, this minor issue did not seem to affect the performance of the device, but seeking a better solution is advised.

4.1.4 Power Source

The average current draw of the device under normal conditions was around 5 A, which is a high current value for a mobile device but still feasible to supply with a commercial RC LiPo battery. The buck converter worked as intended, providing great flexibility for current and future uses.

5 A of current demand is still high for a mobile device

4.1.5 Driver Unit

In order to analyze the dynamic performance of the system, a stress tests was conducted. All vibration motors were turned on and off at regular intervals of 500 ms while monitoring the voltage and the current. A voltage drop of 0.5 V was measured when all motors were active at their operating voltage. The filtering of the motor units and the ICs proved to be sufficient hence no extra decoupling capacitors were needed.

The filtering of the motor units and the ICs proved to be sufficient

4.1.6 Enclosure

The location of the enclosure did not interfere with normal movements of the users. Although the enclosure was made as small as possible, its size was still noticeable. Future iterations could further reduce the size of the enclosure to make it more inconspicuous.

4.1.7 Vest

The type of fabric proved to be a good choice for the vest. It provided a tight and comfortable fit and adapted to users with different body sizes thanks to its elasticity and the different velcro strips.

4.1.8 Code

An in-depth analysis of every section of the main loop reveals that the performance of the function `getFrame()`, responsible for capturing and resizing the depth frame, acts as a bottleneck for the rest of the code. The runtime of the main loop with only this particular function stays below 110 ms, or 9 fps. The function `setVibration()`, in charge of setting the PWM values obtained with the function `getFrame()`, drops the frame rate even more, obtaining an average frame rate for the complete program of 4 fps.

The performance of OpenCV using Python is as good as using C++

The Python version of OpenCV is a wrapper for the original C/C++ code, combining best features of both the languages: the performance of C/C++ and the simplicity of Python. When a function in OpenCV is called from Python, the underlying C/C++ source is run, basically obtaining the same performance as if the code was written on C/C++ .

The function `getFrame()` uses mainly OpenCV functions, obtaining the best performance possible, therefore this poor performance should theoretically be attributed to the performance of the SBC and not to the code.

4.2 User Study

The time constraints set for the completion of this Bachelor Thesis did not allow for the formulation of a formal user study with BVI users. In order to properly evaluate the performance of OpenVNAVI as well as to address potential unanticipated issues identified by the final users of this device, it is highly advised to carry out a thoroughly planned user study following the completion of this Bachelor Thesis.

4.3 Fab Lab

The Fab Lab was key for the construction of the prototype

The OpenVNAVI prototype was developed in its entirety in Aachen, provided by the Media Computing Group. The availability of Fast Prototyping machines such as Fused Deposition Modeling (FDM) 3D printers, a laser cutter, a PCB milling machine, as well as a plethora of tools and mechanical and electronic components, was paramount for the creation of this prototype.

Figure 4.1: Fab Lab Aachen

Chapter 5

Summary and Future Work

In this chapter the author summarizes the results of this Bachelor Thesis and performs an exploration on the features that could be implemented to further improve the capabilities of OpenVNAVI.

5.1 Summary and Contributions

Obstacle avoidance is possible for obstacles large enough to survive the downsampling process

The time constraints set for the completion of this Bachelor Thesis allowed the implementation of only part of the great number of possibilities that the current hardware is able to offer. The main goal of this thesis, obstacle avoidance, could only be achieved for obstacles large enough to survive the downsampling process and for those within the sensing capabilities of the sensor and the human body. For smaller objects, a different technique is later proposed. Turn-by-turn navigation was left out of this implementation for reasons of complexity.

OpenVNAVI provides better resolution than the previous state of the art in this category

Regarding the contributions, the author designed and implemented a flexible platform with higher actuator resolution than the previous state of the art, enabling the possibility for perform analysis on BVI users. Preliminary tests show promising results but further studies should be performed to correctly assess the capabilities and limitations of this device.

Figure 5.1: User wearing OpenVNAVI

5.2 Future Work

Apart from a user study, sever performance improvements and features could be implemented to further improve OpenVNAVI.

5.2.1 Performance Improvements

The issue of detecting obstacles at close proximity needs to be addressed

The depth sensing technology of the ASUS Xtion PRO LIVE sensor is susceptible to interferences from sensors of the same kind. To avoid interferences between users wearing the OpenVNAVI, sensors based on Time of Flight technologies that allow for the modification of the modulation frequency could be used. The lack of sensing capabilities for distances shorter than 50 cm for long-range sensors could be addressed by attaching one or two ultrasonic sensors to warn about an impending collision.

As discussed in the previous chapter, the performance of the current implementation in terms of frame rate is sufficient for testing but it should be improved in future iterations. The suggested solution is to use a more powerful SBC.

The system should be more power efficient

A substantial reduction in power consumption is needed to achieve full mobility and all-day battery life. A possible solution would be to use LRAs instead of ERM vibration motors, possibly reducing the current consumption by 50%, at the expense of increasing the cost of the device.

Regarding the footprint of the hardware, the design of the Driver PCB could be made into a Raspberry Pi shield to make it more compact and convenient. Technological advancements in the next years would allow for a complete integration of the sensor and electronics with the vest.

5.2.2 Text-to-Speech and Feature Detection

According to the 2007 Annual Report from the American Printing House for the Blind, only 10% of U.S. children use Braille as their primary reading medium [American Printing House for the Blind, 2007]. Text-to-Speech seems to be more convenient for most applications and most BVI users, for instance, are not able to read, or even find, braille information on signs placed on the wall.

The RGB camera of the sensor could be used for feature detection

Currently there are smartphone apps that use text-to-speech technology to allow BVI users to read that information. With the current hardware, it could be possible to implement those features in OpenVNAVI. Some relevant features like stairs, doors or even faces could also be detected in order to further improve the interaction experience of the user with the environment.

5.2.3 Turn-by-Turn Navigation

Outdoor GPS-based turn-by-turn navigation is now possible thanks to the availability of many smartphone apps that serve that purpose. On the other hand, indoors navigation has been, and still is a challenge. With no GPS coverage, researchers over the last decades have tried using RFID and Bluetooth beacons, but their high cost and the impracticality of a global deployment have deemed them inviable.

Project Tango could be used to expand the capabilities of OpenVNAVi in the future

Google's Project Tango platform is showing promising results in regards to indoors mapping and is currently capable of offering indoors turn-by-turn navigation thanks to its area learning features. A global deployment of this technology similar to what Google Street View now offers could offer indoors turn-by-turn navigation for both sighted and BVI users.

A practical application of this technology on OpenVNAVI could interact with the user by using pulsating vibration on the leftmost and rightmost columns of actuators of the array, indicating when to turn, with the frequency of the pulses proportional to the angle error.

5.2.4 Floor-level Obstacle Detection

Small obstacles at floor level such as steps or holes cannot be detected with the current implementation of OpenVNAVI. The main contributing factors are the limitations in sensitivity of the depth sensor, the level at which the human skin is able to discriminate different vibration values and also the loss of information that occurs in the downsampling process.

Thanks to the high number of vibration motor units that compose OpenVNAVI, different features could be implemented in different areas of the array. To solve the problem at hand, the author proposes a solution based on computer vision algorithms.

First, using both RGB and depth information from the sensor, a detection a segmentation of the floor plane is performed. Then, analyzing the disparities of color and depth, a possible obstacle is detected and assigned to one of the 16 vibration motors of the last row. Using vibrational cues such as pulsating vibration, the user could be alerted of a possible danger ahead.

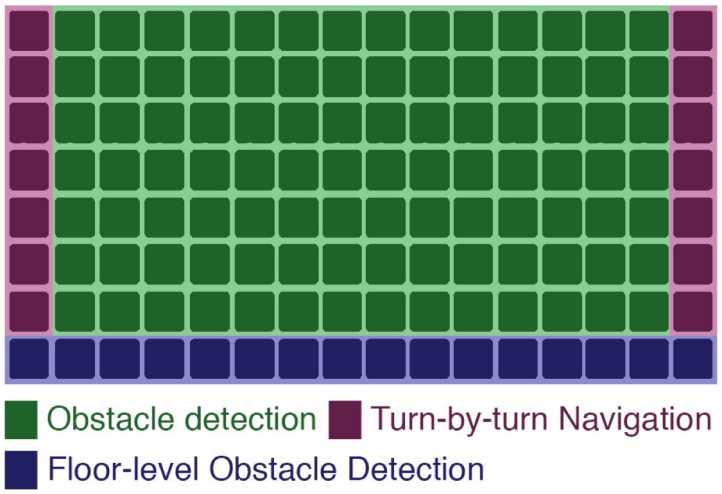


Figure 5.2: Functional areas of the array

Appendix A

Code

OpenVNAVI GitHub Repository[a]

[a] https://github.com/davidanton/OpenVNAVI

```
# ========================================
# Imports
# ========================================
import cv2
import numpy as np
import sys
import os
import time
import RPi.GPIO as GPIO
from PWM_driver import PWM

# ========================================
# Constants
# ========================================
""" Each device and channel is identified
   by an integer. The C++ version of OpenCV
    defines some constants for the
   identifiers of certain devices and
   channels. However, these constants are
   not defined in the Python version and
   the assignment needs to be performed
```

```
    manually. """
16
17 # Devices.
18 CV_CAP_OPENNI = 900 # OpenNI (for Microsoft
     Kinect)
19 CV_CAP_OPENNI_ASUS = 910 # OpenNI (for Asus
     Xtion)
20 # Channels of an OpenNI-compatible depth
    generator.
21 CV_CAP_OPENNI_DEPTH_MAP = 0 # Depth values
    in mm (CV_16UC1)
22 CV_CAP_OPENNI_POINT_CLOUD_MAP = 1 # XYZ in
    meters (CV_32FC3)
23 CV_CAP_OPENNI_DISPARITY_MAP = 2 # Disparity
     in pixels (CV_8UC1)
24 CV_CAP_OPENNI_DISPARITY_MAP_32F = 3 #
    Disparity in pixels (CV_32FC1)
25 CV_CAP_OPENNI_VALID_DEPTH_MASK = 4 #
    CV_8UC1
26 # Channels of an OpenNI-compatible RGB
    image generator.
27 CV_CAP_OPENNI_BGR_IMAGE = 5
28 CV_CAP_OPENNI_GRAY_IMAGE = 6
29
30 # ========================================
31 # Definitions
32 # ========================================
33 def depthTest():
34
35     """ Grabs frames from different
    channels, multiplies the output by
    different constant gain values and saves
     the PNG frames for visual inspection of
     the range of depth values of a scene.
    The folder named "test_frames" must be
    created. """
36
37     abc = (["a","b","c","d","e","f","g","h"
    ,"i","j","k","l","m","n","o","p","q","r"
    ,"s","t","u","v","w","x","y","z"])
38     values = 7
39     testGain = 5
40     maxChannel = 4
```

```
41      for i in range(0, values):
42          for j in range(0, maxChannel):
43              capture.grab()
44              print "Retrieving channel " +
    str(j) + abc[i]
45              success, rawFrame = capture.
    retrieve(channel = j)
46              frame640j = testGain * i *
    rawFrame
47              cv2.imwrite("test_frames/" +
    str(j) + abc[i] + ".png", frame640j)
48
49  def sweepTest():
50
51      """ Sweeps up and down for all motors.
    """
52
53      minPWM = 300
54      maxPWM = 3000
55      step = 100
56      for i in range(minPWM,maxPWM,step):
57          for row in range(0,8):
58              for col in range(0,16):
59                  IC[row].setPWM(col, 0, i)
60                  time.sleep(0.001)
61                  print ("IC: " + str(row) +
    " motor: " +
62                         str(col) + " PWM: " +
     str(i))
63
64      for i in range(maxPWM,minPWM,-step):
65          for row in range(0,8):
66              for col in range(0,16):
67                  IC[row].setPWM(col, 0, i)
68                  time.sleep(0.001)
69                  print ("IC: " + str(row) +
    " motor: " +
70                         str(col) + " PWM: " +
     str(i))
71
72  def strobeTest(row):
73
```

```
74      """ Strobes maximum and minimum values
        for a given row. """
75
76      minPWM = 300
77      maxPWM = 3000
78      IC[row].setAllPWM(0, minPWM)
79      time.sleep (1)
80      IC[row].setAllPWM(0, maxPWM)
81      time.sleep (1)
82
83  def fadeIn():
84
85      """ Ramps from low to medium vibration
        values for a smooth
86      transition. """
87
88      minPWM = 300
89      maxPWM = 1000
90      step = 10
91      for i in range(minPWM,maxPWM,step):
92          for row in range(0,8):
93              IC[row].setAllPWM(0, i)
94
95  def fadeOut():
96
97      """ Ramps from medium to low vibration
        values for a smooth
98      transition. """
99
100     minPWM = 300
101     maxPWM = 1000
102     step = 10
103     for i in range(maxPWM, minPWM, -step):
104         for row in range(0,8):
105             IC[row].setAllPWM(0, i)
106
107 def beep(number, t):
108
109     """ Simple buzzer beep. Takes number of
        beeps and the duration. """
110
111     GPIO.setwarnings(False)
112     GPIO.setup(12, GPIO.OUT)
```

```
113     for i in range (0, number):
114         buzzer = GPIO.PWM(12, 5000)
115         time.sleep(t)
116         buzzer.start(92)
117         time.sleep(t)
118         buzzer.stop()
119 
120 
121 def pause():
122 
123     """ Pause function. """
124 
125     fadeOut()
126     for row in range(0, 8):
127         IC[row].setAllPWM(0, 0)
128     print "System paused by the user."
129     time.sleep(0.5)
130     beep(1, 0.2)
131     print "System ready, press switch to
    continue..."
132     flag = 1
133     GPIO.wait_for_edge(18, GPIO.RISING)
134     fadeIn()
135 
136 def getFrame():
137 
138     """ Gets the frame from the sensor and
    stores it it.
139     Returns 16x8 numpy array. """
140 
141     capture.grab()
142     success, rawFrame = capture.retrieve(
    channel = channel)
143     frame640 = gain * rawFrame
144     # Black border removal
145     height = 480
146     width = 640
147     frame640_crop = frame640[15:height-5,
    12:width-22]
148 
149     frame16 = cv2.resize(frame640, (16, 8))
150     frame16 = frame16.astype(int)
151     # For debugging
```

```
152     # cv2.imwrite("frame640.png", frame640)
153     # cv2.imwrite("frame16.png", frame16)
154     # print frame16
155     return frame16
156 
157 def setVibration():
158 
159     """ Sets PWM values to each vibration
    motor unit. """
160 
161     mappingFunction = 8 # Mapping of
    grayscale values to PWM values.
162     PWM16 = mappingFunction * getFrame()
163     maxPWM = 3060
164     for row in range(0,8):
165         for col in range (0,16):
166             if PWM16[row,col] > maxPWM:
167                 PWM16[row,col] = maxPWM
168                 IC[row].setPWM(col,0,(PWM16
    [row,col]))
169             elif PWM16[row,col] == 0:
170                 PWM16[row,col] = maxPWM
171                 IC[row].setPWM(col,0,(PWM16
    [row,col]))
172             else:
173                 IC[row].setPWM(col,0,(PWM16
    [row,col]))
174                 # print (PWM16[row,col]),
175         # print "\n"
176 
177 # ========================================
178 # Main function
179 # ========================================
180 
181 def main():
182 
183     try:
184         # Initialization of PWM drivers.
    PWM(0x40, debug=True) for debugging.
185         global IC
186         IC = []
187         freq = 490
188         for i in range(0,8):
```

```
189             IC.append(PWM(0x40+i))
190             IC[i].setPWMFreq(freq)
191 
192         # Initialization of the sensor.
193         global sensor
194         sensor = CV_CAP_OPENNI_ASUS
195         global channel
196         channel = 3
197         global gain
198         gain = 5
199         global capture
200         capture = cv2.VideoCapture(sensor)
201         capture.open(sensor)
202         while not capture.isOpened():
203             print "Couldn't open sensor. Is
     it connected?"
204             time.sleep(100)
205         print "Sensor opened successfully"
206 
207         # GPIO initialization.
208         GPIO.setmode(GPIO.BCM)
209         GPIO.setup(18, GPIO.IN,
    pull_up_down=GPIO.PUD_DOWN)
210         sw1 = GPIO.input(18) # Input NO
    switch.
211         flag = 0 # Flag for start/stop
    button.
212 
213         # Waits for sw1 to be pressed.
214         print "System ready, press switch
    to continue..."
215         beep(1, 0.2)
216         GPIO.wait_for_edge(18, GPIO.RISING)
217         fadeIn()
218 
219         while True:
220             if ((GPIO.input(18) == False)
    or
221                 (GPIO.input(18) == True and
     flag == 1)):
222                 flag = 0
223                 tick = time.clock()
224 
```

```
225                 # depthTest()
226                 # sweepTest()
227                 # strobeTest(0)
228                 # getFrame()
229                 setVibration()
230 
231                 tock = time.clock()
232                 runtime = tock - tick
233                 print "Loop runtime: " +
    str(runtime) + "s"
234                 print "FPS: " + str(int(1/
    runtime))
235             else:
236                 pause()
237 
238     except KeyboardInterrupt:
239         GPIO.cleanup
240         for i in range(0, 8):
241             IC[i].setAllPWM(0, 0)
242         print "Shutdown requested. \
    nExiting..."
243 
244     finally:
245         print "Done"
246 
247 if __name__ == "__main__":
248     main()
```

Appendix B

Installing OpenCV and OpenNI

Updated version available on:

OpenVNAVI GitHub Repository[a]

[a] https://github.com/davidanton/OpenVNAVI

B.1 Installing OpenCV 2.4.3

Before building OpenCV, a fresh install of Raspbian is required. The following code snippets must be copied and pasted on the Terminal.

B.1.1 Installing Dependencies

Update:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get rpi-update
```

Developer tools and packages:

```
sudo apt-get -y install build-essential
   cmake pkg-config
```

QT and GTK development libraries:

```
sudo apt-get -y install libqt4-dev libgtk2
   .0-dev
```

Image I/O packages:

```
sudo apt-get -y install libjpeg8-dev
   libtiff4-dev libjasper-dev libpng12-dev
   libpng12-0 libpng++-dev libpng3
   libpnglite-dev libjasper-dev libjasper-
   runtime libjasper1 pngtools libtiff4-dev
    libtiff4 libtiffxx0c2 libtiff-tools
   libjpeg8 libjpeg8-dev libjpeg8-dbg
   libjpeg-progs libjpeg-dev
```

Video video I/O packages:

```
sudo apt-get -y install libavcodec-dev
   libavformat-dev libswscale-dev libv4l-
   dev libavcodec53 libavformat53
   libavutil51 libavutil-dev libswscale2
   libswscale-dev libgstreamer0.10-0-dbg
   libgstreamer0.10-0 libgstreamer0.10-dev
   libxine1-ffmpeg libxine-dev libxine1-bin
    libunicap2 libunicap2-dev libgstreamer-
   plugins-base0.10-dev
```

Other libraries:

```
sudo apt-get -y install libatlas-base-dev
   gfortran swig zlib1g-dbg zlib1g zlib1g-
   dev
```

```
sudo apt-get -y install libpython2.7 python
   -dev python2.7-dev python-numpy python-
   pip
```

B.1.2 Compiling OpenCV

Download:

```
cd /home/pi/
wget https://github.com/Itseez/opencv/
   archive/2.4.3.tar.gz
tar -xzf opencv-2.4.3.tar.gz
cd opencv-2.4.3
mkdir build
cd build
```

Configuration:

```
sudo cmake -D CMAKE_BUILD_TYPE=RELEASE -D
   CMAKE_INSTALL_PREFIX=/usr/local -D
   BUILD_NEW_PYTHON_SUPPORT=ON -D
   BUILD_EXAMPLES=ON -D WITH_OPENNI=YES ..
```

Compilation and installation:

```
sudo make
sudo make install
```

B.1.3 Post-installation Setup

```
sudo echo /usr/local/lib > /etc/ld.so.conf.
   d/opencv.conf
sudo echo PKG_CONFIG_PATH=$PKG_CONFIG_PATH
   :/usr/local/lib/pkgconfig >> /etc/bash.
   bashrc
sudo echo export PKG_CONFIG_PATH >> /etc/
   bash.bashrc
```

B.2 Installing OpenNI 1.5.8.5

```
sudo apt-get -y install libusb-1.0 doxygen
   freeglut3-dev openjdk-6-jdk graphviz
```

```
cd /home/pi/
git clone https://github.com/OpenNI/OpenNI.
   git -b unstable
git clone https://github.com/PrimeSense/
   Sensor.git -b unstable
```

```
sudo nano OpenNI/Platform/Linux/Build/
   Common/Platform.Arm
```

Replace this:

```
CFLAGS += -march=armv7-a -mtune=cortex-a8 -
   mfpu=neon -mfloat-abi=softfp #-mcpu=
   cortex-a8
```

With this:

```
CFLAGS += -mtune=arm1176jzf-s -mfpu=vfp -
   mfloat-abi=hard
```

```
cd /home/pi/OpenNI/Platform/Linux/
   CreateRedist/
./RedistMaker.Arm
cd /home/pi/OpenNI/Platform/Linux/Redist/
   OpenNI-Bin-Dev-Linux-Arm-v1.5.8.5
sudo ./install.sh
```

```
sudo nano /home/pi/Sensor/Platform/Linux/
   Build/Common/Platform.Arm
```

Replace this:

```
make -j$(calc_jobs_number) -C ../Build
```

With this:

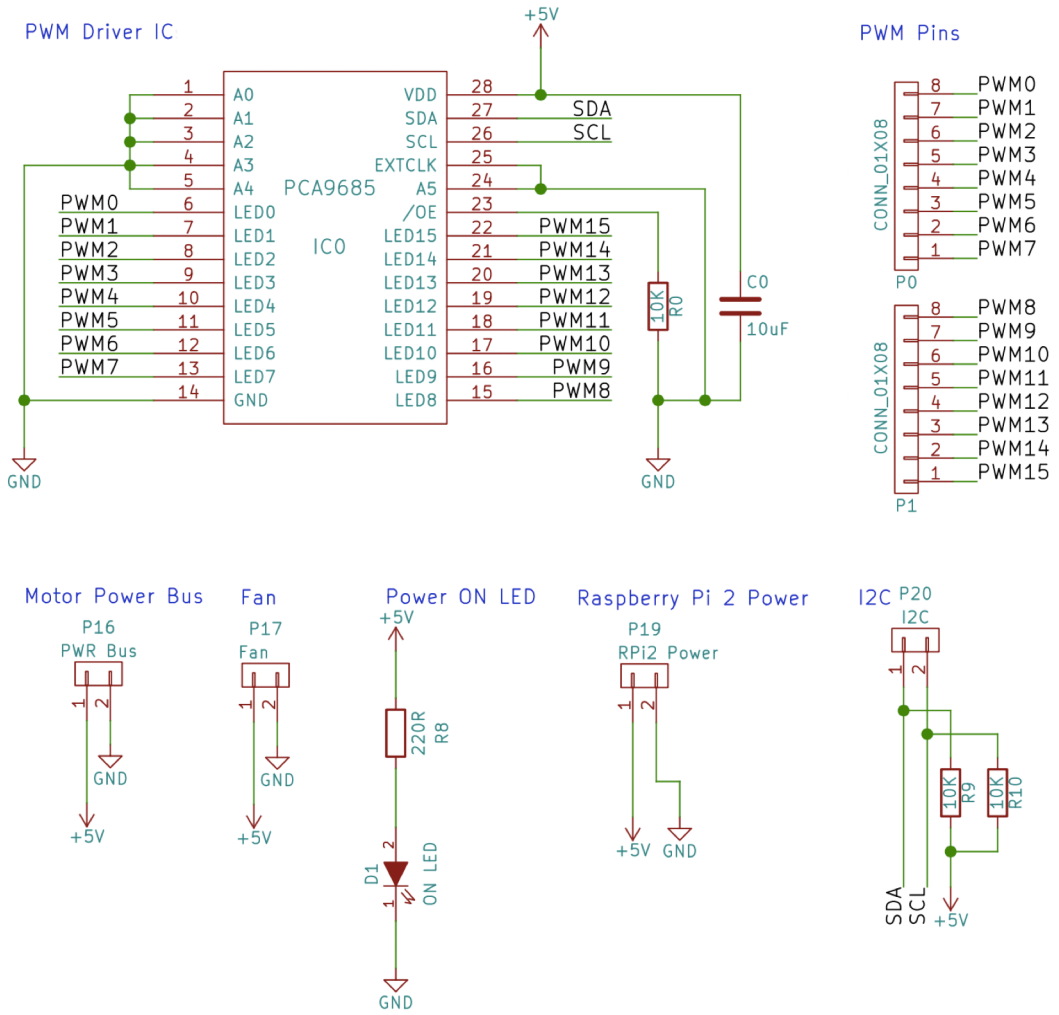
```
make -j1 -C ../Build
```

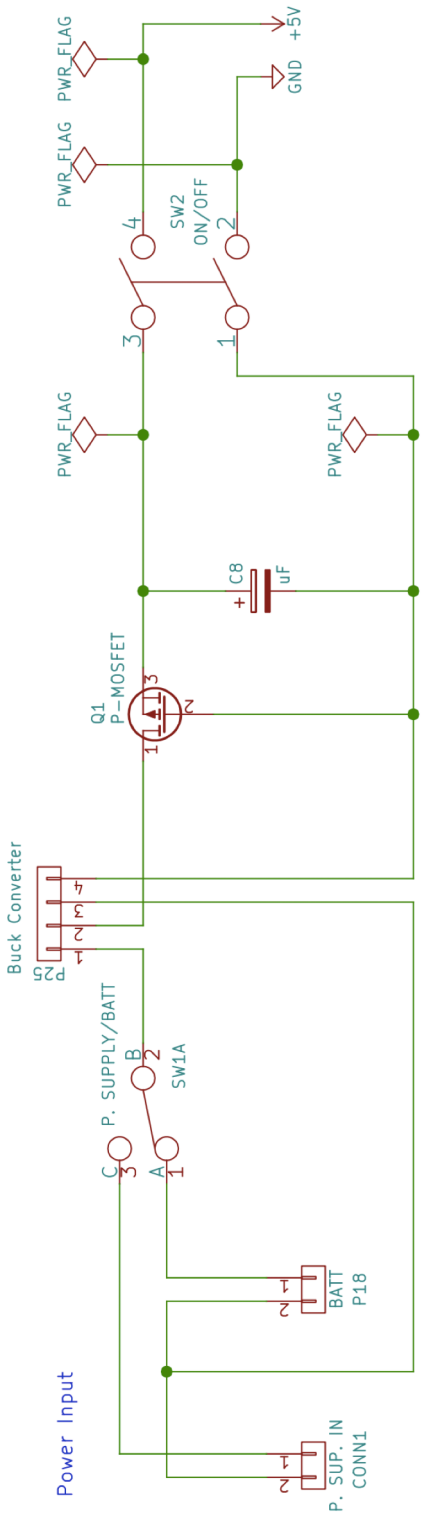
```
cd /home/pi/Sensor/Platform/Linux/
   CreateRedist/
./RedistMaker Arm
```

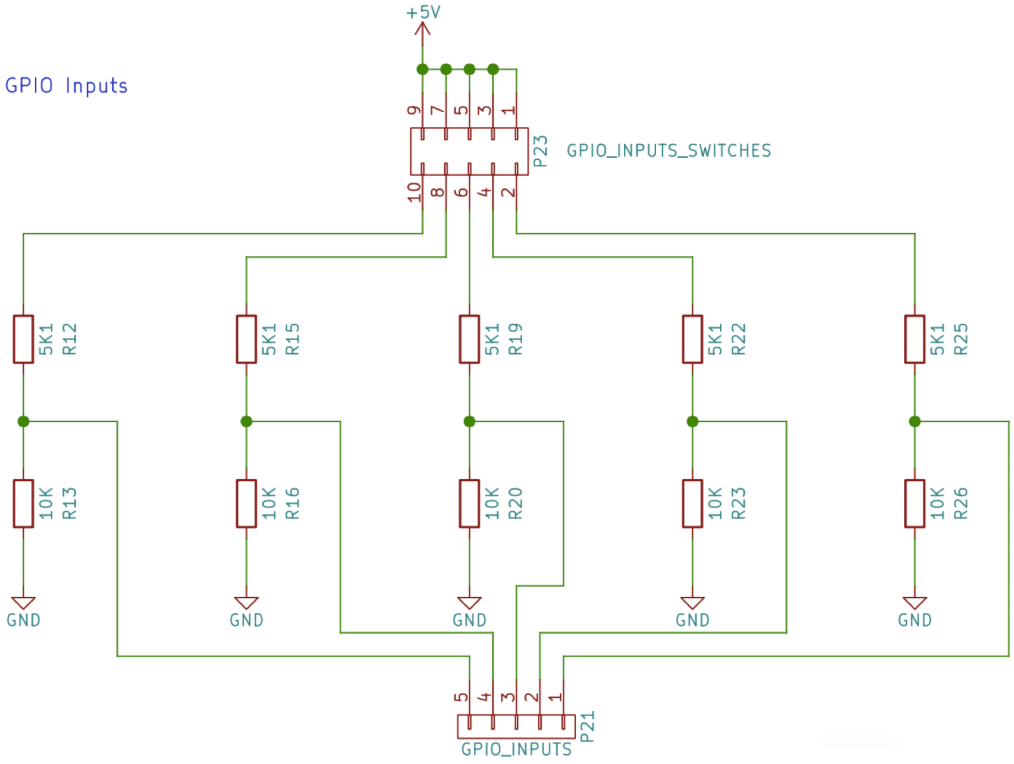
```
cd /home/pi/Sensor/Platform/Linux/Redist/
   Sensor-Bin-Linux-Arm-v5.1.6.5
sudo ./install.sh
```

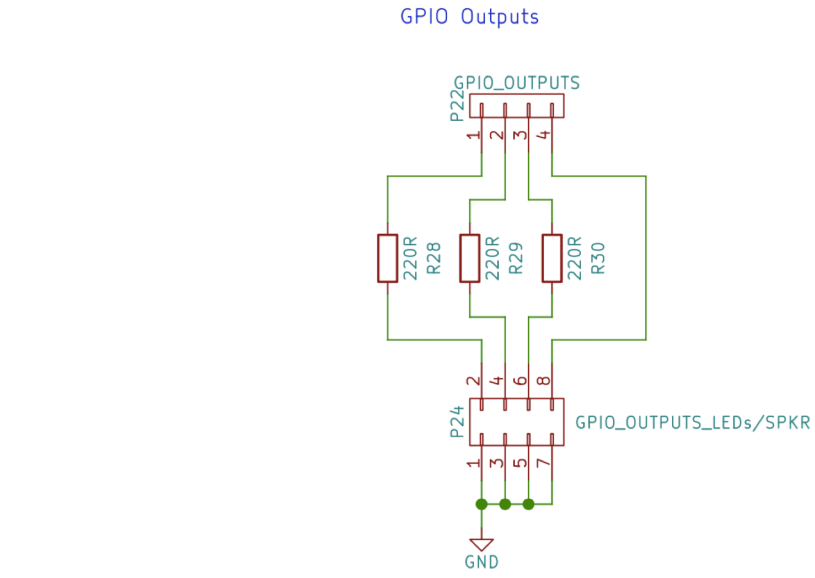
Appendix C

Driver Unit Schematics


PWM Driver IC
PCA9685
IC0
A0
A1
A2
A3
A4
LED0
LED1
LED2
LED3
LED4
LED5
LED6
LED7
GND
VDD
SDA
SCL
EXTCLK
A5
/OE
LED15
LED14
LED13
LED12
LED11
LED10
LED9
LED8
+5V
PWM0
PWM1
PWM2
PWM3
PWM4
PWM5
PWM6
PWM7
PWM8
PWM9
PWM10
PWM11
PWM12
PWM13
PWM14
PWM15
10K
R0
C0
10uF
PWM Pins
CONN_01X08
P0
P1
Motor Power Bus
P16
PWR Bus
Fan
P17
Power ON LED
220R
R8
D1
ON LED
Raspberry Pi 2 Power
P19
RPi2 Power
I2C P20
I2C
10K
R9
10K
R10
SDA
SCL


PWR_FLAG
PWR_FLAG
+5V
GND
SW2
ON/OFF
4
2
3
1
PWR_FLAG
PWR_FLAG
C8
uF
Q1
P-MOSFET
Buck Converter
P2S
P. SUPPLY/BATT
SW1A
B
C
A
BATT
P18
Power Input
P. SUP. IN
CONN1


GPIO Inputs
+5V
P23
GPIO_INPUTS_SWITCHES
5K1 R12
5K1 R15
5K1 R19
5K1 R22
5K1 R25
10K R13
10K R16
10K R20
10K R23
10K R26
GND
P21
GPIO_INPUTS


GPIO Outputs
P22
GPIO_OUTPUTS
220R R28
220R R29
220R R30
P24
GPIO_OUTPUTS_LEDs/SPKR
GND

Appendix D

Making the Vest

The materials required for making the vest are a piece of 1.3x1.5 m dark gray 2% elastane 98% cotton fabric, dark gray thread and 9 m of velcro.

Figure D.1: Fabric, thread and velcro

16 velcro strips of 36 cm in length spaced 3 cm from each other are sewn to the fabric using a zigzag pattern.

Figure D.2: Sewing of the velcro strips

Fabric strips are placed vertically in-between the velcro strips to serve as cable guides. Other cable guides were also placed on the interior of the vest to guide the bundle cables over the shoulders from the actuators to the enclosure.

Figure D.3: Velcro strips and cable guides

Horizontal velcro strips on the exterior allow the vest to adapt to different chest, waist and hip measurement sizes.

Figure D.4: Vest exterior

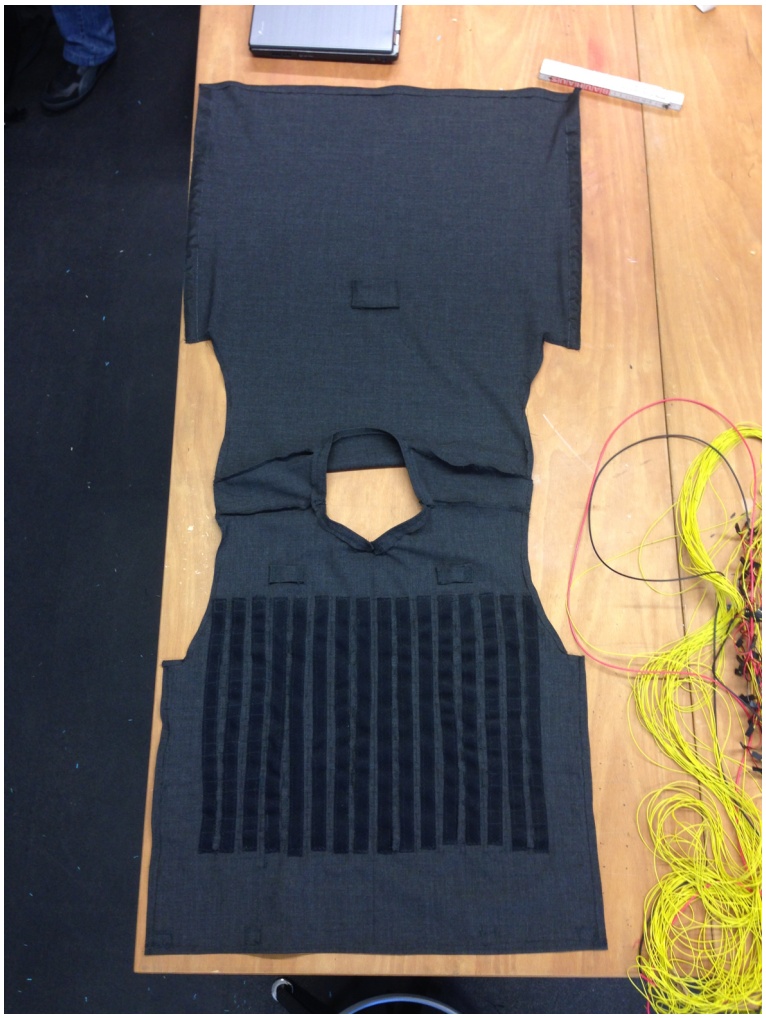


Figure D.5: Vest interior

The 3-way female pin headers are used to connect 5 V, ground and the PWM signal to the motor units.

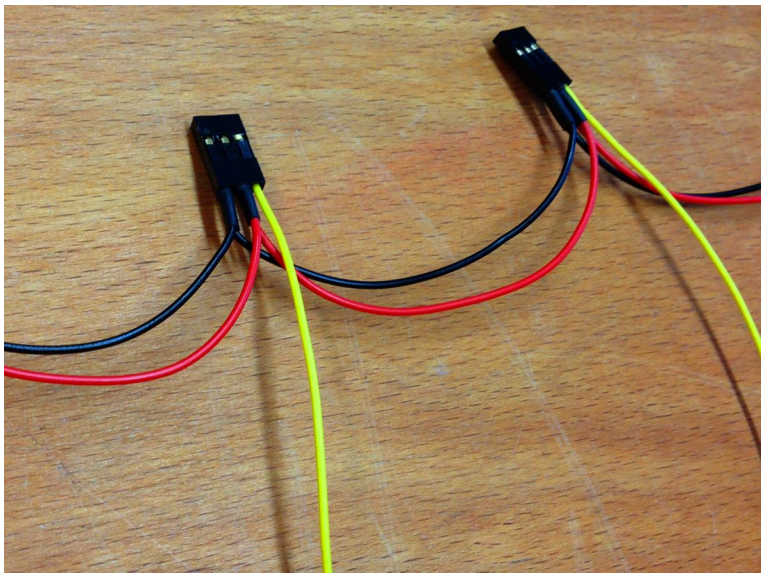


Figure D.6: Motor Unit cabling

Figure D.7: Cabling fo the vest

The 5 V and ground cables coming from each row are respectively bound together using a 2-way screw terminal at the bottom of the array. This reduces the number of power cables that need to from the array to the enclosure to only two.

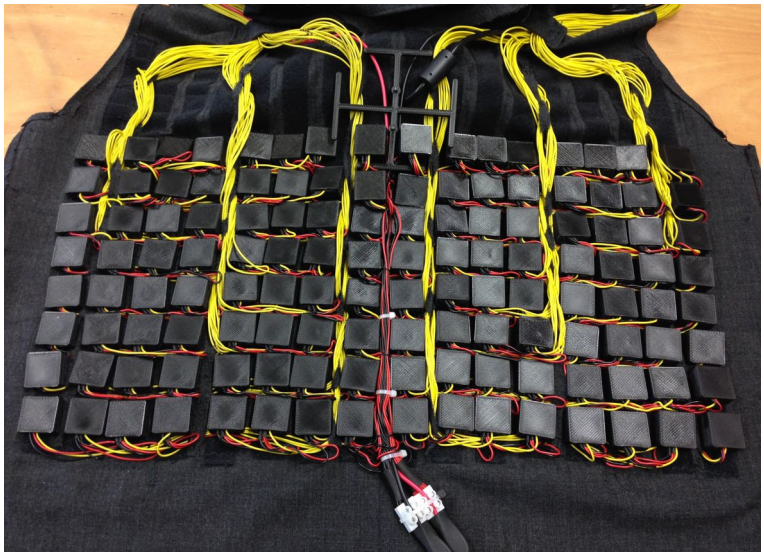


Figure D.8: Vibration motor array

Figure D.9: Motor Unit enclosure

Figure D.10: OpenVNAVI (Front)

Figure D.11: OpenVNAVI (Back)

Bibliography

3D Haptic Vest by Sean Benson. Online, accessed: June 15th, 2015. URL https://hackaday.io/project/1962-3d-haptic-vest-for-visually-impaired-and-gamers.

Fab Lab Aachen. URL http://hci.rwth-aachen.de/fablab/.

Agencia Brasil. CC Antonio Cruz. Online, accessed: Aug 5th, 2015. URL http://agenciabrasil.ebc.com.br/.

American Printing House for the Blind. Facts and Figures on Americans with Vision Loss, 2007.

Paul Bach-y Rita. Tactile sensory substitution studies. *Annals of the New York Academy of Sciences*, 1013(1):83–91, 2004. ISSN 1749-6632. doi: 10.1196/annals.1305.006. URL http://dx.doi.org/10.1196/annals.1305.006.

J.C. Bliss, M.H. Katcher, C.H. Rogers, and R.P. Shepard. Optical-to-tactile image conversion for the blind. *Man-Machine Systems, IEEE Transactions on*, 11(1):58–65, March 1970. ISSN 0536-1540. doi: 10.1109/TMMS.1970.299963.

Timo Breuer, Christoph Bodensteiner, and Michael Arens. Low-cost commodity depth sensor comparison and accuracy analysis. volume 9250, pages 92500G–92500G–10, 2014. doi: 10.1117/12.2067155. URL http://dx.doi.org/10.1117/12.2067155.

Sylvain Cardin, Daniel Thalmann, and Frédéric Vexo. A wearable system for mobility improvement of visually impaired people. *The Visual Computer*, 23:109–118, 2007. doi: http://dx.doi.org/10.1007/s00371-006-0032-4.

D. Dakopoulos, S.K. Boddhu, and N. Bourbakis. A 2d vibration array as an assistive device for visually impaired. In *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, pages 930–937, Oct 2007. doi: 10.1109/BIBE.2007.4375670.

Eyeronman. Online, accessed: June 15th, 2015. URL http://www.tactilenavigationtools.com/eyeronman/.

Frank A. Geldard. Some neglected possibilities of communication. *The Journal of the Acoustical Society of America*, 131:1583–1588, 1960. doi: http://dx.doi.org/10.1126/science.131.3413.1583.

Frank A. Geldard and Carl E. Sherrick. Multiple cutaneous stimulation: The discrimination of vibratory patterns. *The Journal of the Acoustical Society of America*, 37(5), 1965.

F. Gemperle, N. Ota, and Dan Siewiorek. Design of a wearable tactile display. In *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, pages 5–12, 2001. doi: 10.1109/ISWC.2001.962082.

Guide Dogs UK Charity for the Blind and Partially Sighted. Online, accessed: Aug 5th, 2015. URL http://www.guidedogs.org.uk/.

J. Hakkinen, T. Vuori, and M. Paakka. Postural stability and sickness symptoms after hmd use. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, volume 1, pages 147–152, Oct 2002. doi: 10.1109/ICSMC.2002.1167964.

Joseph Howse, editor. *OpenCV Computer Vision with Python*. Packt Publishing, 2013. ISBN 978-1782163923. URL https://www.packtpub.com/application-development/opencv-computer-vision-python.

L.A. Johnson and C.M. Higgins. A navigation aid for the blind using tactile-visual sensory substitution. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 6289–6292, Aug 2006a. doi: 10.1109/IEMBS.2006.259473.

L.A. Johnson and C.M. Higgins. A navigation aid for the blind using tactile-visual sensory substitution. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 6289–6292, Aug 2006b. doi: 10.1109/IEMBS.2006.259473.

L.A. Jones, M. Nakamura, and B. Lockyer. Development of a tactile vest. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th International Symposium on*, pages 82–89, March 2004. doi: 10.1109/HAPTIC.2004.1287181.

Lynette A. Jones, Brett Lockyer, and Erin Piateski. Tactile display and vibrotactile pattern recognition on the torso. *Advanced Robotics*, 20(12):1359–1374, 2006. doi: 10.1163/156855306778960563. URL http://dx.doi.org/10.1163/156855306778960563.

K.A. Kaczmarek. The tongue display unit (tdu) for electrotactile spatiotemporal pattern presentation. *Scientia Iranica*, 18(6):1476 – 1485, 2011. ISSN 1026-3098. doi: http://dx.doi.org/10.1016/j.scient.2011.08.020. URL http://www.sciencedirect.com/science/article/pii/S1026309811001702.

Kurt A. Kaczmarek and Paul Bach-Y-Rita. Virtual environments and advanced interface design. pages 349–414, New York, NY, USA, 1995. Oxford University Press, Inc. ISBN 0-19-507555-2. URL http://dl.acm.org/citation.cfm?id=216164.216182.

Eric Richard Kandel, James Harris Schwartz, Thomas M. Jessell, and Sarah Mack, editors. *Principles of Neural Science*. McGraw-Hill Medical, New York, Chicago, San Francisco, 2013. ISBN 978-0-07-139011-8. URL http://opac.inria.fr/record=b1135227.

L. Kay. Auditory perception of objects by blind persons, using a bioacoustic high resolution air sonar. *J.Acoust.Soc.Am.*, 2000.

Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437, 2012. ISSN 1424-8220. doi: 10.3390/s120201437. URL http://www.mdpi.com/1424-8220/12/2/1437.

Kinect for the Blind. Online, accessed: June 15th, 2015. URL http://www.zoneos.com/kinectfortheblind.htm.

Kinecthesia. Online, accessed: June 15th, 2015. URL http://www.kinecthesia.com/.

Ig Mo Koo, Kwangmok Jung, Ja Choon Koo, Jae-Do Nam, Young Kwan Lee, and Hyouk Ryeol Choi. Development of soft-actuator-based wearable tactile display. *Robotics, IEEE Transactions on*, 24(3):549–558, June 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.921561.

Eugene C. Lechelt. Sensory-substitution systems for the sensorily impaired: The case for the use of tactile-vibratory stimulation. *Perceptual and Motor Skills*, 62:356–358, 1985. doi: 10.2466/pms.1986.62.2.356.

Kurniawan S Manduchi, R. Mobility-related accidents experienced by people with visual impairment. *Insight: Research and Practice in Visual Impairment and Blindness*, 4(2), 2011. URL http://users.soe.ucsc.edu/~manduchi/papers/MobilityAccidents.pdf.

Steve Mann, Jason Huang, Ryan Janzen, Raymond Lo, Valmiki Rampersad, Alexander Chen, and Taqveer Doha. Blind navigation with a wearable range camera and vibrotactile helmet. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1325–1328, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0616-4. doi: 10.1145/2072298.2072005. URL http://doi.acm.org/10.1145/2072298.2072005.

Precision Microdrives. Online, accessed: May 4th, 2015. URL http://www.precisionmicrodrives.com/.

Saskia K Nagel, Christine Carl, Tobias Kringe, Robert Märtin, and Peter König. Beyond sensory substitution—learning the sixth sense. *Journal of Neural Engineering*, 2(4):R13, 2005. URL http://stacks.iop.org/1741-2552/2/i=4/a=R02.

G. Ng, P. Barralon, G. Dumont, S.K.W. Schwarz, and J.M. Ansermino. Optimizing the tactile display of physiological information: Vibro-tactile vs. electro-tactile stimulation, and forearm or wrist location. In *Engineering in*

Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pages 4202–4205, Aug 2007. doi: 10.1109/IEMBS.2007.4353263.

I2C Bus Specification and User Manual. NXP.

G. Pineda Garcia, J. Ortiz Bejar, and F. Mercado Miramontes. A prototype helping device for the visually impaired using an optical to mechanical transducer. In *Power, Electronics and Computing (ROPEC), 2013 IEEE International Autumn Meeting on*, pages 1–4, Nov 2013. doi: 10.1109/ROPEC.2013.6702721.

Raspberry Pi Foundation. Raspberry Pi, 2008. Online, accessed: June 3rd, 2015. URL https://www.raspberrypi.org/.

N. Reyes-Ayala, N. Felix-Gonzalez, and H. Perez-Ponce. A secondary aid that helps to create cognitive maps for visual impaired persons. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 2, pages 1484–1487 vol.2, 2001. doi: 10.1109/IEMBS.2001.1020486.

Frank A. Saunders. Information transmission across the skin: High-resolution tactile sensory aids for the deaf and the blind. *International Journal of Neuroscience*, 19(1-4):21–28, 1983. doi: 10.3109/00207458309148642. URL http://informahealthcare.com/doi/abs/10.3109/00207458309148642.

Brandon T. Shrewsbury. Providing haptic feedback using the kinect. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, pages 321–322, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0920-2. doi: 10.1145/2049536.2049628. URL http://doi.acm.org/10.1145/2049536.2049628.

Shutterstock. Image ID: 78856573. Online, accessed: Aug 5th, 2015. URL http://www.shutterstock.com.

R. Tapu, B. Mocanu, and E. Tapu. A survey on wearable devices used to assist the visual impaired user navigation in outdoor environments. In *Electronics and Telecommunications (ISETC), 2014 11th International Symposium on*, pages 1–4, Nov 2014. doi: 10.1109/ISETC.2014.7010793.

Time-of-Flight Camera - An Introduction. Texas Instruments.

Koji Tsukada and Michiaki Yasumura. Activebelt: Belt-type wearable tactile display for directional navigation. In Nigel Davies, ElizabethD. Mynatt, and Itiro Siio, editors, *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 384–399. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22955-1. doi: 10.1007/978-3-540-30119-6_23. URL http://dx.doi.org/10.1007/978-3-540-30119-6_23.

Hendrik A.H.C van Veen and Jan B.F. van Erp. Providing directional information with tactile torso displays. 2003. URL http://www.eurohaptics.vision.ee.ethz.ch/2003/71.pdf.

R. Velazquez, E.E. Pissaloux, J.C. Guinot, and F. Maingreaud. Walking using touch: Design and preliminary prototype of a non-invasive eta for the visually impaired. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 6821–6824, Jan 2005. doi: 10.1109/IEMBS.2005.1616071.

R. Velazquez, O. Bazan, and M. Magaa. A shoe-integrated tactile display for directional navigation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1235–1240, Oct 2009. doi: 10.1109/IROS.2009.5354802.

Ramiro Velázquez. Brisben, a. j. and hsiao, s. s. and johnson, k. o. *Annals of the New York Academy of Sciences*, 81, 1999.

Ramiro Velázquez. Wearable and autonomous biomedical devices and systems for smart environment. *Annals of the New York Academy of Sciences*, 75:331–349, 2010. doi: 10.1007/978-3-642-15687-8_17. URL http://dx.doi.org/10.1007/978-3-642-15687-8_17.

viSparsh. Online, accessed: June 15th, 2015. URL http://visparsh.blogspot.de/.

BenjaminW. White, FrankA. Saunders, Lawrence Scadden, Paul Bach-Y-Rita, and CarterC. Collins. Seeing with the skin. *Perception and Psychophysics*, 7(1):23–27, 1970. URL http://dx.doi.org/10.3758/BF03210126.

B.W. White. Perceptual findings with the vision-substitution system. *Man-Machine Systems, IEEE Transactions on*, 11(1):54–58, March 1970. ISSN 0536-1540. doi: 10.1109/TMMS.1970.299962.

Lorraine Whitmarsh, editor. *The Benefits Of Guide Dog Ownership*. 2005.

World Health Organization. Online, accessed: June 5th, 2015. URL http://www.who.int/en/.

Juan Wu, Zhenzhong Song, Weixiong Wu, Aiguo Song, and D. Constantinescu. A vibro-tactile system for image contour display. In *VR Innovation (ISVRI), 2011 IEEE International Symposium on*, pages 145–150, March 2011. doi: 10.1109/ISVRI.2011.5759619.

JoséR. Álvarez Sánchez, Félix de la Paz, and José Mira. A robotics inspired method of modeling accessible open space to help blind people in the orientation and traveling tasks. 2005. URL http://dx.doi.org/10.1007/11499220_42.

Typeset August 22, 2015