The Design of a Wearable Tactile Array Which can Exploit the Funneling Illusion in Two Dimensions

by

Kacey Cayen

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science (MSc) in Computational Science

The Faculty of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne

Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	The Design of a Wearable Tactile Array Which can Exploit the Funneling Illusion in Two Dimensions
Name of Candidate Nom du candidat	Cayen, Kacey
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	March 18, 2021

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Ratvinder Grewal
(Supervisor/Directeur(trice) de thèse)

Dr. Glenn Legault
(Committee member/Membre du comité)

Dr. Kalpdrum Passi
(Committee member/Membre du comité)

Dr. Pradeep Atrey
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. Lace Marie Brogden
Madame Lace Marie Brogden
Acting Dean, Faculty of Graduate Studies
Doyenne intérimaire, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

The goal of this research was to create a wearable tactile array that is capable of exploiting the funneling illusion in two dimensions. The design process is explored in detail, including the formulation of the design requirements, the materials that were trialed, the electronics design iterations, and the development of a software development kit (SDK) which includes a novel algorithm for rendering both the static funneling illusion and dynamic funneling illusion. The ability of the algorithm to render overlapping circle and square stimuli at discriminable levels is experimentally tested demonstrating its efficacy in this task.

Keywords

Design, Wearables, Tactile Array, Wearable Tactile Array, Funneling Illusion, Two Dimensional Funneling Illusion Algorithm, Software Design.

Acknowledgments

I would foremost like to thank my supervisor, Dr. Ratvinder Grewal, whose patience and support as I explored various avenues for this project is greatly appreciated. I would also like to thank Stephen Podrucky for designing and assembling the PCBs for this project.

Table of Contents

List of Tables

List of Figures

Figure 12: A histogram of motor current draws at 3.0V from 74 motors.

Figure 13: Early cloth prototype with tensor bandages.

Figure 14: Early cloth prototype worn around the waist.

Figure 15: A 3D printed modular plastic prototype.

Figure 16: The 1st 3D printed digital rubber prototype.

Figure 17: The 2nd 3D printed digital rubber prototype.

Figure 18: Low density open cell foam prototype.

Figure 19: A foam panel with attached tactors for the wearable tactile array.

Figure 20: The garment for the wearable tactile array.

Figure 21: A breadboard circuit for controlling four motors.

Figure 22: A rubber panel prototype.

Figure 23: The 1st version of the modular PCBs along with it's 3D printed plastic housings.

Figure 24: The 2nd version of the modular PCBs.

Figure 25: The 3rd version of the modular PCBs.

Figure 26: The back side of a foam panel with attached modular PCBs in plastic housings.

List of Appendices

1.0 Introduction

People's lives have been enhanced by technology that allows information to be displayed to them; we have speakers which confer audible information and televisions which confer visual information. Skin has been suggested as a modality for communicating information for over 50 years (Geldard, 1957). The springboard for research into this area was the Symposium on Cutaneous Sensitivity in 1960 which brought together the top researchers in this area including Frank Geldard and Georg von Békésy (Hawkes, 1960; Gallace, et. al, 2007). Since that time, technology has improved to the point that conferring information through the tactile modality is feasible with off the shelf parts.

One of the best designs for conferring tactile information is the tactile array, a grid of tactors which can be activated in defined patterns that presents some information to the user. The funneling illusion is a tactile illusion in which a user can be made to feel a point of stimulus in between tactors which allows a sparse grid of tactors to act as a much denser grid. Utilizing this illusion, the perceived point of stimulus can also be moved over the surface of the user's skin within the tactile array allowing for even greater variety of information to be conferred to the user.

The objective of this thesis is therefore to create a wearable tactile array, that is one that a user can comfortably wear, that is capable of exploiting the funneling illusion in two dimensions. To best achieve our objective we will take a look at the literature to help determine the design

requirements for the wearable tactile array as well as give context to the work on a novel two dimensional funneling illusion rendering algorithm, this will be section 2. In section 3, we will dive into the design of the soft goods, electronics, software, and two dimensional funneling illusion rendering algorithm that goes into making the wearable tactile array. Section 4 will cover the experiment methodology used in an experiment which demonstrates the effectiveness of the two dimensional funneling illusion rendering algorithm. In section 5, the results of the experiment will be presented and discussed. And finally, section 6 will conclude this thesis and discuss further work that can be done.

2.0 Literature Review

In this section we will give a brief overview of human cutaneous receptors (section 2.1), receptive fields (section 2.2), and the funneling illusion (section 2.3). We will then take a deeper dive into haptics with a focus on how previous work has helped shape the design requirements for the wearable tactile array. We will look at previous work on haptic devices (section 2.4),, and haptics exploiting the funneling illusion (section 2.5). We will conclude this section with a review of previous work on building design guidelines for wearables and tactile arrays (section 2.6).

2.1 Cutaneous Receptors

In order to best design a wearable tactile display, it is important to have an understanding on how the user perceives tactile stimuli.

Humans have four main receptor groups; the cutaneous and subcutaneous mechanoreceptors for touch perception (Johnson, 2001), the thermal receptors for temperature perception (Schepers & Ringkamp, 2010), the nociceptors for pain perception (Munger & Ide, 1988), and the muscle and skeletal mechanoreceptors for limb proprioception (Michelson & Hutchins, 1995). Even though there are multiple ways to induce stimulation of thermoreceptors; these include contact heat, indirect thermal heat by focused light bulb, and a variety of lasers (argon, copper vapour, semiconductor, neodymium-YAG, thulium-YAG, and CO2) (Arendt-Nielsen & Chen, 2003), the focus of this thesis will be on stimulating mechanoreceptors.

There are four main mechanoreceptors; the Meissner corpuscle for stroking and flutter perception, the Merkel disk receptor for pressure and texture perception, the Ruffini ending for skin stretch perception, and the Pacinian corpuscle for vibration perception (Johnson, 2001).

2.2 Receptive Fields

A receptive field in its simplest form is an area of stimulus space over which changes in the stimulus cause corresponding increases or decreases in the firing rate of the neuron. This definition works well for the cutaneous mechanoreceptors but receptive fields get substantially more complex as you move up the processing layers.

The layers of the tactile perception pathway look like this (Hsiao et al, 2009):

1. The four afferents originating from the cutaneous receptors; slowly adapting type 1 (SA1), slowly adapting type 2 (SA2), rapidly adapting (RA), and Pacinian (PC).

2. The dorsal column-medial lemniscus (DCML) pathway which is composed of three groupings of neurons.

    a. First-order neurons which are sensory neurons located in the dorsal root ganglia that send their afferent fibers through the two dorsal columns (the gracile fasciculus and the cuneate fasciculus).

    b. The first-order axons make contact with the second-order neurons at the gracile nucleus and the cuneate nucleus in the lower medulla.

    c. These neurons send their axons to the third-order neurons in the ventral nuclear group of the thalamus.

3. The next stop from there is the primary somatosensory cortex.

4. Then continuing on to the secondary somatosensory cortex.

5. The pathway beyond the secondary somatosensory cortex is not well defined.

Each layer has their own receptive field shapes with cutaneous afferents having simple central surround structures (i.e. a central region of excitation with surround suppression) while cortical receptive fields are more complex with spatiotemporal traits with a central excitatory region surrounded by one or more regions of inhibition (Hsiao et al, 2009).

2.3 Funneling Illusion

In order to get better performance from the wearable tactile array, we can exploit a tactile illusion. The funneling illusion, first described by Békésy in 1958 is a phenomenon that is produced when two vibrational actuators are activated on the surface of the skin in which the subject will perceive a single point stimulus in the space between the actuators. The location of the perceived stimulus is dependent upon the difference in intensities of the two actuators. If the actuators are of equal magnitude then the stimulus is perceived as being in the middle of the two actuators. If the intensity of one of the actuators is higher than the other, the stimulus is perceived as being closer to the actuator with the more intense activation. By exploiting this characteristic, the perceived stimulus can be moved along the axis in between two actuators by varying the intensities of the actuators over time.

By exploiting the funneling illusion we can use a sparser array and achieve similar results to a denser array with the added bonus of being able to render dynamic tactile stimuli. The funneling illusion can be invoked in thermoreceptors as well, which was demonstrated using peltier elements (contact heat) on the forearm of subjects (Oohara et al., 2010). The receptive fields of the first order neurons encode stimulus accurately to the position of the mechanical stimulators (Gardener & Spencer, 1972) demonstrating that the funneling illusion must be caused by sensory integration further in the pathway. This was also demonstrated by Kuroki et al (2007) who created an electromechanical stimulator which was composed of a 3 x 5 grid of 1.25mm diameter electrodes with a 2.5mm spacing driven by a linear actuator. This device was attached to the

users finger. The goal of the researchers was to look at the effects of the interaction between the electrical and mechanical stimuli on perception of the resulting stimuli. Since electrical stimulation with the electrode is unable to stimulate the deeper Pacinian Corpuscles, the researchers hypothesized that the interaction would be different if the stimulator was driven at 40Hz, targeting Meissner Corpuscles or at 240Hz, targeting the Pacinian Corpuscles. The addition of the mechanical stimulation reduced the perceptual "electric feeling" of the electrocutaneous stimulation. Their results demonstrated that the interaction effect remained the same, indicating that the perceptual quality of the stimulus is integrated in the central nervous system. This is further demonstrated by Chen et al. (2003) who were able to demonstrate using intrinsic-signal optical imaging that the funneling illusion produces discrete, focal activations in area 3b of the somatosensory cortex that are located between the two single digit activations in anesthetized squirrel monkeys. Blankenburg et al. (2006) were able to demonstrate utilizing fMRI with human participants that tactile perceptual illusions generate the same somatosensory activation as actual tactile stimuli (i.e. if the same area of the forearm is stimulated by an actual motor or by a tactile illusion utilizing motors on each side of the area, the resulting somatosensory activation is the same). Therefore, we should be able to utilize any of the mechanoreceptors to exploit this illusion.

Pacinian corpuscles have large, non defined receptive fields with a single zone of maximal sensitivity with a continuous decrease in sensitivity as you move outward from this zone (Johansson, 1978) and Sherrick et al (1990) demonstrated that a 250Hz stimuli is capable of stimulating the Pacinian receptors without stimulating the non-Pacinian receptors. These

properties of the Pacinian corpuscle make it the ideal candidate for the target of our wearable tactile array.

2.4 Haptics

The Merriam-Webster dictionary defines haptic as “relating to or based on the sense of touch.” What this means when it comes to technology development is that haptics should further the user experience through the exploitation of their tactile modality. This can be done in many ways. This section will explore the previous work in building haptic devices (section 2.4.1) and tactile array (section 2.4.2) to help inform our design requirements in the wearable tactile array design section of this thesis. Previous work will help in the decision for which tactors to use for the wearable tactile array as well as influence our design choices in order to enable flexible tactile information design. The studies described in this section will be tagged with the design requirement they helped inform.

A detailed overview of how the design requirements were chosen is covered in section 3.1. The twenty requirements that were chosen are as follows:

1. The placement of the wearable tactile array should be on a large surface area with limited movement and flexibility.

2. The wearable tactile array should feel safe and comfortable on the user.

3. The wearable tactile array should not protrude away from the users body such that it interferes with their natural movements and ability to move through the environment.

4. The wearable tactile array should fit various body sizes.

5. The wearable tactile array should be light enough that it does not encumber the user.

6. The tactile signals should be of sufficient magnitude that they are attended but not aversive.

7. The wearable tactile array should be aesthetically pleasing.

8. The wearable tactile array should be relatively silent.

9. The wearable tactile array should have low power consumption so it can be untethered if necessary.

10. The wearable tactile array should be easy to put on and take off.

11. The components of the wearable tactile array should be cost effective.

12. The tactors should be small.

13. The intertactor spacing should stay consistent even when the user moves.

14. The tactors should be held tight to the body.

15. The tactors should maintain contact with the user when the user moves.

16. The vibration from each tactor should be isolated to that tactor.

17. The system should be integratable into existing systems.

18. The system should be easily configurable.

19. The system should be expandable.

20. The delay in the system should be less than 25 ms.

2.4.1 General Haptics

It has been demonstrated that spatially-predictive vibrotactile cues can rapidly shift visual attention. Ho et al (2005) used two vibrotactile actuators, one positioned in the middle of the participants stomach and the other in the middle of the participants back, held onto the participant with a velcro belt overtop their clothing to spatially cue the participant to imminent collision during a driving task. Participants had faster reaction times when validly spatially-cued demonstrating that a directional tactile cue can shift visual attention to the front or back. Kennett et al (2002) demonstrated that a directional tactile cue can also shift visual attention to the left or right. **Design requirements 16 & 18.**

Dinh et al (1999) created an experiment in which two specific tactile cues were provided; as the user passed in front of the virtual fan, a real fan was turned on to blow on the user, and when the user steps onto the balcony a heat lamp is activated. They demonstrated that the addition of tactile cues to a virtual environment increased the users sense of presence as well as their ability to remember object locations in the environment. **Design requirements 5, 9 & 18.**

Many haptic devices have been created for a variety of research purposes. One team created a wireless ring shaped device with an embedded eccentric rotating mass (ERM) motor which

makes direct skin contact to give the users proximity-based cues using different vibrotactile patterns (Ariza et al., 2015). While Cockburn and Brewster (2005) demonstrated that adding a vibrotactile cue increased small-target acquisition in graphical user interfaces. **Design requirements 12 & 18.**

Hein and Brell (2007) created a haptic glove for use with computer aided surgery. The glove was composed of 4 vibrating motors on the back of the hand along with a positional tracking system. The researchers utilized pulsed vibration at one of the four motors at a time to guide the users to a desired hand position in two dimensional space with the pulses becoming more intense as the users got closer to the desired position. **Design requirements 2, 3, 4, 5, 6, 8, 9, 12, 14, 15, 16, 17, 18 & 20.**

Hummel et al (2016) designed a novel electrotactile feedback device because they felt that current haptic feedback devices were typically uncomfortable and hard to integrate into immersive virtual reality (VR) systems. The device was composed of five electrotactile boards measuring 6mm x 18mm which were secured to the palmar side of the distal phalanxes with a plastic thimble which had embedded LEDs for finger tracking. These boards were connected to generation boards in plastic housings worn around the forearm. Using the electrotactile feedback device in an immersive virtual environment, users showed faster completion times and more precision in performing a button press task, a lever switch task, and an on-orbit servicing scenario task. **Design requirements 2, 3, 5, 9, 12, 14, 15, 17 & 18.**

Muramatsu et al (2012) developed a haptic glove as a “easy-to-use, easily portable, and lightweight” interface for interacting with virtual objects. Their glove was composed of five coin

type ERM vibrating motors positioned on the fingernails of each finger and thumb along with five flexible bending sensors to detect finger and thumb movement. The researchers used this glove to test users subjective impressions of vibrotactile stimuli. **Design requirements 5, 9, 10, 12, 14, 15, 17 & 18.**

Spagnoletti et al (2018) developed a wearable fingertip haptic stimulator to provide users with the sensations of touching objects made of different materials as well as providing pressure and texture rendering. It is composed of five main elements; a small servo motor housed within the upper body which is located on the nail side of the finger, the mobile end-effector which is in contact with the finger pad, the three cables which attaches the two previously mentioned parts together allowing the platform to move towards or away from the fingertip, a voice coil actuator enabling the device to provide vibrotactile stimuli for texture rendering, and a clip which allows for the fastening of the device on the finger. The researchers performed one experiment to determine the ability of users to distinguish between three textures using the device in a virtual environment. The device varied the pressure provided to the fingertip by moving the platform away or towards the fingertip to provide the interaction forces from the contact of the fingertip with the virtual environment. Rendering of textures was done through the use of the Penn Haptic Texture Toolkit which includes 100 haptic texture and friction models but only three of these textures were used. The experiment started by presenting subjects with three virtual objects made of different materials; a red object made of felt, a pale brown object made of aluminium, and a dark brown object made of brick. The users had two minutes in this virtual environment to familiarize themselves with the textures of the three objects. The subjects were then presented with a different environment with a black object and had to determine its material. The ten

subjects were able to determine the proper material with a 63±15% accuracy. **Design requirements 5, 9, 12, 14, 15, 17 & 18.**

Kim and Colgate (2012) developed a multi-functional tactor that can display touch, pressure, vibration, shear force, and temperature to the skin. The goal of this study was to see if this type of tactor can enhance control of grip force during amputees' prosthesis operation during a grip-and-lift task that calls for minimum pressure. This grip-and-lift task was performed using a virtual prosthetic arm to manipulate a virtual object. Although the tactor was multi-functional, only the pressure and shear force stimulations were used in the experiment. The users demonstrated lower grip pressure when haptic feedback in the form of shear or pressure was provided. Although having both shear and pressure feedback simultaneously had lower grip pressure than no haptic feedback, it had higher grip pressure than both individual haptic feedback conditions. Kim and Colgate (2012) suspected that the more natural cognitive mapping in their targeted muscle reinnervation (TMR) patients would lead to better haptic feedback, although Moriyama et al (2018) demonstrated that the use of a haptic pressure device on the forearm to simulate the pressure felt at the thumb and index fingers in a virtual environment had users indicate that they felt it to be a natural mapping. **Design requirement 2.**

Perceptual illusions such as the rubber-hand illusion which induces the participant to perceive being touched on a fake arm as though it was part of their body and the elongated-arm illusion which extends the body space of the participant with elongated virtual limbs can be induced in immersive virtual environments. The rubber-hand illusion can be extended to the entire body in what is called the body-transfer illusion. These illusions can be induced with vibrotactile haptic

feedback glove (Ariza et al., 2016). The vibrotactile glove that the researchers designed is composed of 14 linear resonant actuators (LRAs). There is one actuator on the thumb, two on each finger, two on the ball of the hand, two on the palmar surface, and one right below the index finger. The motors are attached to a thin fabric glove with rubber bands. The glove uses Bluetooth Low Energy (BLE) to communicate with the computer. The intensity and location of stimulation was determined by what the researchers termed Tactile Control Points (TCPs). These TCPs were colliders implemented in the virtual environment which were located on the avatars hands at the correlating points of the motors. The TCPs intensity value was calculated by a distance function defined by $\left(1 - \frac{x}{0.01}\right)^4$ which returns a value greater than zero for distances less than a centimeter. The total latency in the system from hand tracking to motor activation was calculated to be 43ms. This latency is suboptimal as Jay et al (2007) demonstrated that delayed haptic feedback above 25ms can negatively affect task performance. **Design requirements 5, 9, 12, 14, 15, 17, 18 & 20.**

2.4.2 Tactile Arrays

Tactile arrays are larger, more complex haptic systems that allow for more information transfer to occur between the system and the user. We shall now explore some of the previous research done with these devices.

Debus et al (2004) created a handheld vibrotactile display to aid in teleoperated assembly. The device consisted of four floating arrays of pins, each pin array consisting of 4 x 11 pins spaced 2.5mm apart, sitting atop of four piezoelectric bimorph beams. These were embedded into a cylindrical housing having a length of 100mm and a diameter of 32mm with a mass of 130

grams. The device is held with a power grasp (as one would hold a hammer), with the palm, thumbs, index, and middle finger in contact with the pin arrays. The use of the vibrotactile device reduced the peak insertion force in a teleoperated peg insertion task. **Design requirement 12.**

Giannopoulos et al (2012) developed a haptic glove to allow users to determine the shape of a virtual object. The glove was composed of 14 10mm diameter coin type vibrating motors. These motors were mounted on perfoboards and glued to cycling gloves. There were two motors per finger (one on the distal phalanx and one on the proximal phalanx), two on the thumb, and four on the palm (one on the thenar eminence, one on the hypothenar eminence, and two along the proximal palmar region). The user's hand was tracked using a Kinect and collisions with the virtual object was done using an n x n raycasting grid originating at the users point of view and passing through the center of the virtual representation of the motors. Based on the number of collisions passing through each motor, the intensity of vibration of the corresponding physical motor was determined. The glove had poor performance thanks to a multitude of poor design choices. Firstly, the motors were attached to the outside of the glove which kept the motors from having proper contact with the skin. Secondly, the motors were attached to perfoboards which would decrease their performance because of the added weight. Thirdly, the material of the glove allowed for the propagation of vibrations which interfered with the ability of users to localize the vibrations to a specific motor. **Design requirements 5, 12, 13, & 16.**

Cha et al. (2008a) created a haptic jacket to be used as part of an interpersonal communication system they called HugMe. The jacket had a series of ERM motors on the inner part of the upper

arm and the inner part of the upper chest. They then added a layer of fabric over the motors and wires to prevent direct skin contact. The jacket was interacted with using a 3 degree-of-freedom force feedback device on a live 2.5D render of the remote person wearing the jacket. **Design requirements 5 & 14.**

Gemperle et al (2001) created a wearable tactile display. The tactor used by these researchers in their design was a cylindrical ERM motor which they epoxied into an aluminium tube creating an 7.6mm x 17.8mm cylindrical tactor. The vest was made with heavy weight Lycra with a lightweight mesh material on the inside to form taut pockets for the tactor modules to allow optimal contact with the skin. **Design requirement 14.**

Gunther and O'Modhrain (2002) approached haptics through the lens of music. They created a system to play music for the sense of touch which is composed of a computer controlled bodysuit with thirteen embedded vibrotactile transducers along with a pair of headphones. There was one larger vibrotactile transducer which was a woofer worn on the lower back and twelve smaller vibrotactile transducers composed of a modified 40mm flat speaker with a weight attached to the center enclosed in a plastic case. There were 3 of these smaller transducers along the length of each arm and leg. These transducers were embedded into a loose fitting nylon suit and were secured tightly by the users via attached elastic straps. The haptic composition is created using Protools a MIDI based software which the authors admitted was a painstaking and awkward process. The researchers utilized the cutaneous rabbit illusion to give a sense of movement to the haptic compositions. **Design requirements 1, 2, 4, 5, 14, 15 & 14.**

Several tactile array systems have been developed for use in navigation. Tachi et al (1985) describe a system they developed using an electrocutaneous array worn as a belt around the waist that can help the blind navigate with the help of a robotic guide dog they called MELDOG. On the other hand Tang and Beebe (2006) designed an oral electrotactile display to help the blind navigate. Their device had a grid of electrodes embedded on the top of a mouthpiece which caused it to make contact with the roof of the mouth and a pressure sensor array embedded on the bottom of the mouthpiece for the user to interface with the device using their tongue. The grid was composed of 7 x 7 grid of 700μm diameter dome shaped electrodes with an intertactor distance of 2.54mm. Their system would deliver directional information with a moving line (Figure 1) or a moving arrow pattern (Figure 2). They also tested the users threshold levels for a static square stimulus (Figure 3) and a dynamic square stimulus (Figure 4). The threshold levels for dynamic and static stimuli were found to be the same. There were no statistical differences between the differentiation of the line pattern and the arrow pattern although only five participants were used in the study and the trend was that the participants did better with the line stimuli. The users had better performance for differentiating left-right and right-left directional cues than front-back and back-front directional cues. **Design requirements 1, 2, 3, 4, 5, 6, 8, 9, 10, 13, 14, 15, 16 & 18.**

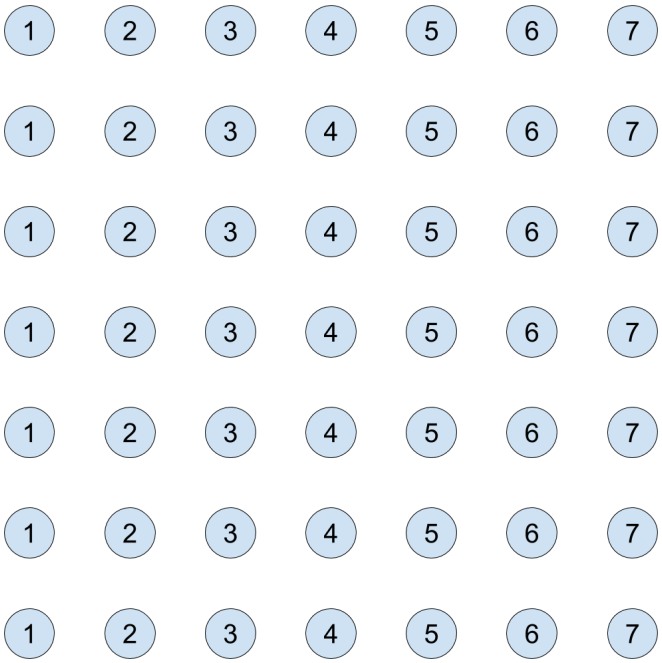


Figure 1. The left to right dynamic line stimulus used in Tang and Beebe (2006).

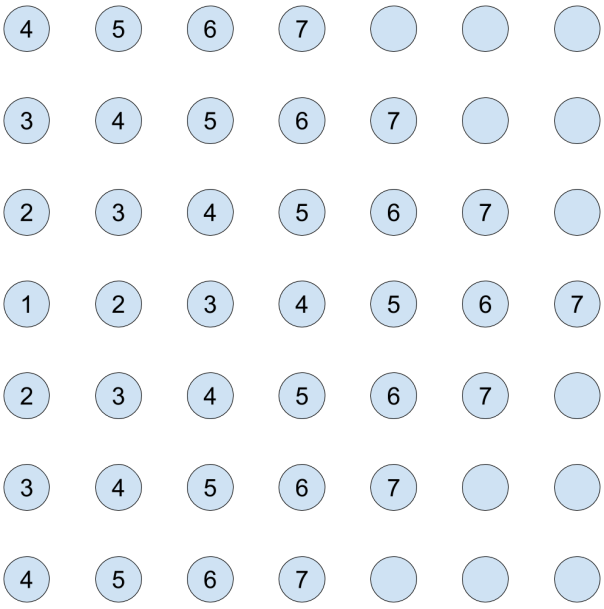


Figure 2. The left to right dynamic arrow stimulus used in Tang and Beebe (2006).

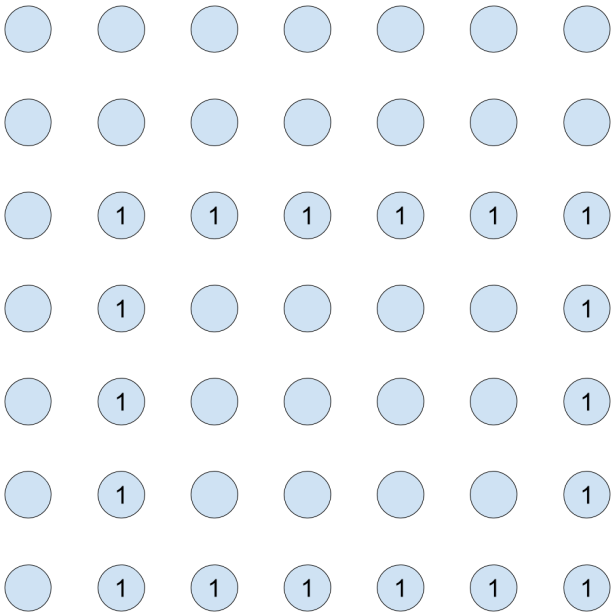


Figure 3. The static square stimulus used in Tang and Beebe (2006).

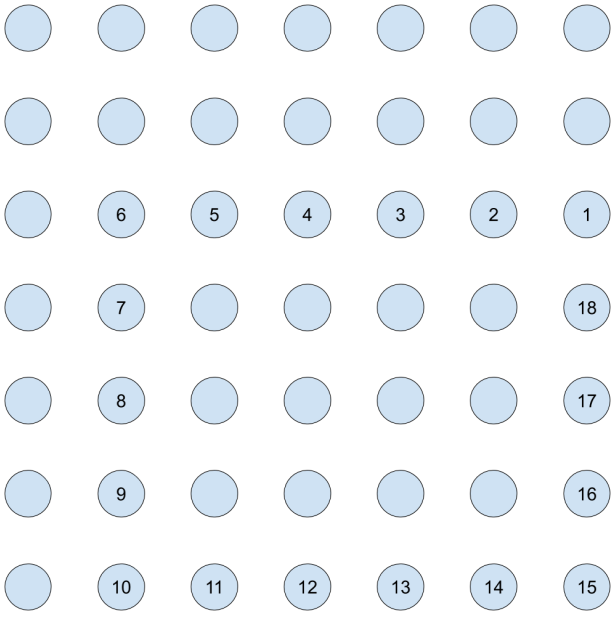


Figure 4. The dynamic square stimulus used in Tang and Beebe (2006).

Piateski and Jones (2005) set out to design a wirelessly controlled, wearable tactile display for use in navigation. To reach the goal they designed some experiments to compare the ability of subjects to identify tactile patterns that were presented either to the forearm or the torso and to determine which tactile patterns are the most easily identified and which body site is the most well suited. Two types of vibrating motors were used: cylindrical motors and coin motors. These two types of motors were chosen because they have a different axis of rotation. The cylindrical motors have an axis of rotation that is perpendicular to the mounting surface while the coin motors have an axis of rotation that is parallel to the mounting surface. Nine motors were used in a 3 x 3 pattern and were glued to a spandex garment to create a tactile display for the forearm with the intermotor spacing being 24mm. Sixteen motors were used in a 4 x 4 pattern and were glued to a spandex garment to create a tactile display for the torso with the vertical intermotor spacing being 40 mm and the horizontal intermotor spacing being 60 mm. There were eight different vibrational patterns presented to the subjects; the forearm patterns can be seen in figure 5, while the torso patterns can be seen in figure 6. The numbers indicate the order of activation of the motors with each activation of motors lasting 500ms with 500ms before subsequent motor activation. Subjects wore headphones with white noise playing during the experiments. In the first experiment, the subjects placed their forearms on a cushion and the motor array was attached. Each pattern was presented to the subjects five times for a total of 40 stimuli, this was done in random order. This was repeated a second time with the alternate motor array. The subjects detected the proper pattern 93.5% of the time with the cylindrical motor array and 85% of the time with the coin motor array. Pattern H was the easiest for the subjects to identify. Patterns that moved across the width of the arm (C, D, and E) were easier for the subjects to

identify than patterns that moved across the length of the arm (A, B, and F). In the second experiment, the subjects were outfitted with the torso garment with the motor array being centered on their back. The same experimental design as experiment one was utilized. The subjects detected patterns C, D, E, F, and H with 100% accuracy, patterns B, and G with 99% accuracy, and pattern A with 97% accuracy. Overall the torso achieved significantly better results as a target for tactile information delivery. **Design requirements 1, 13 & 16.**

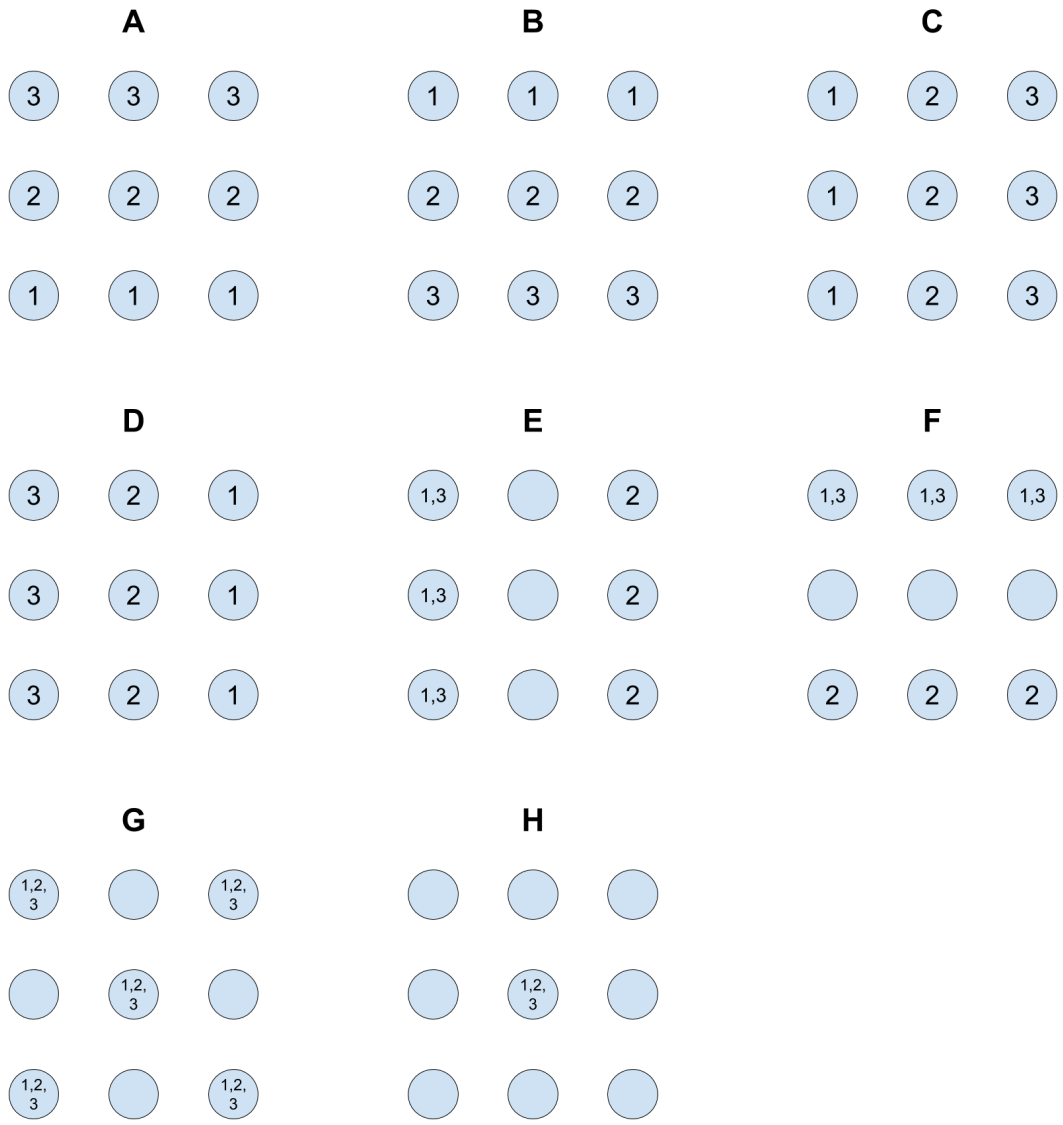


Figure 5. 3 x 3 tactile array patterns used Piateski & Jones (2005).

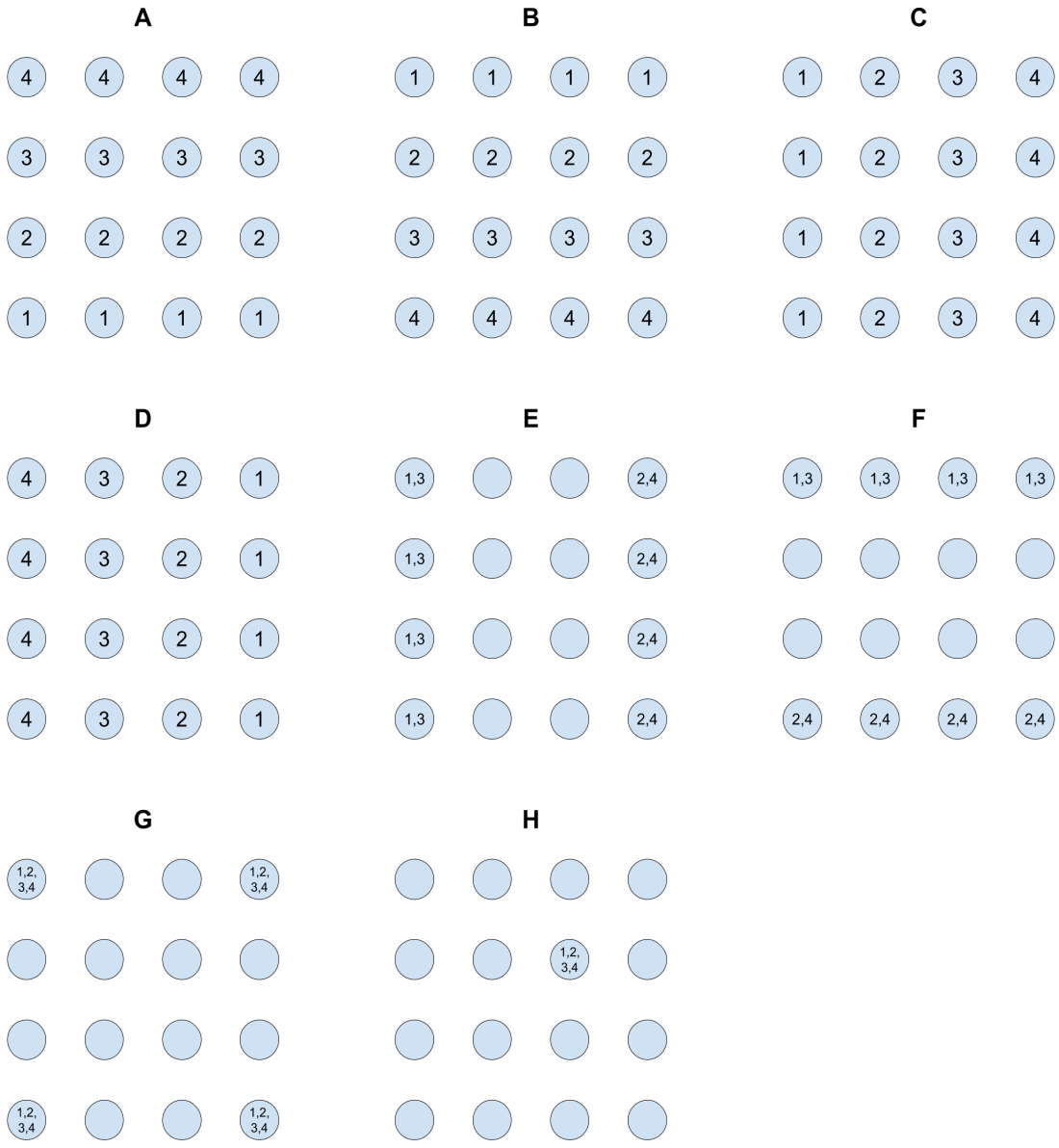


Figure 6. 4 x 4 tactile array patterns used in Piateski & Jones (2005).

Some researchers have developed wearable tactile displays for use in helping pilots navigate their aircrafts. The tactile situation awareness system (TSAS) is a wearable tactile array developed for military aircraft flight control feedback (Rupert et al, 2016). It was demonstrated

that helicopter pilots in a flight simulator were able to maintain a hover in brownout conditions much more reliably with the TSAS then without. Van erp et al (2002) designed a vest composed of custom tactors composed of a DC motor housed in a PVC contactor with a 15mm x 20mm contact area. They embedded 64 of these tactors in a stretch fleece vest with 12 motors around the circumference of the torso and five rows of spaced equidistantly between the navel and the nipples. There were single tactors on each shoulder as well as behind each thigh. Two pulsing rhythm were used to indicate the level of error 100 ms on and 200 ms off for small errors and 50 ms on and 100 ms off for large errors. Apparent movement was done by sequentially activating motors from the spine to the navel in either a left or right direction at two different interstimulus onset intervals to indicate a low speed or a high speed. The use of the vest in a flight simulator task decreased error in both the night vision task and full vision task. The use of apparent motion did not improve performance although their rendering of apparent motion did not exploit either the funneling illusion or the cutaneous rabbit illusion. **Design requirements 1, 2, 3, 4, 5, 6, 9, 10, 12, 13, 17 & 18.**

Yanagida et al (2004) utilized a 3 x 3 matrix of 18mm diameter ERM vibrating motors affixed to the back of an office chair as a tactile array. Utilizing a tracing algorithm they developed, they were able to get subjects to distinguish alphanumeric characters with an 87% accuracy. Their algorithm sequentially activated single motors in the tactile array to trace the letters for the subject (see Figure 7). Although subjects were unable to distinguish between a “Z” character and a “2” character because their system could not produce curves. They were not able to utilize the funneling illusion because of the limitations of the ERM motors used in their array. **Design requirements 12, 13, 14, 16 & 18.**

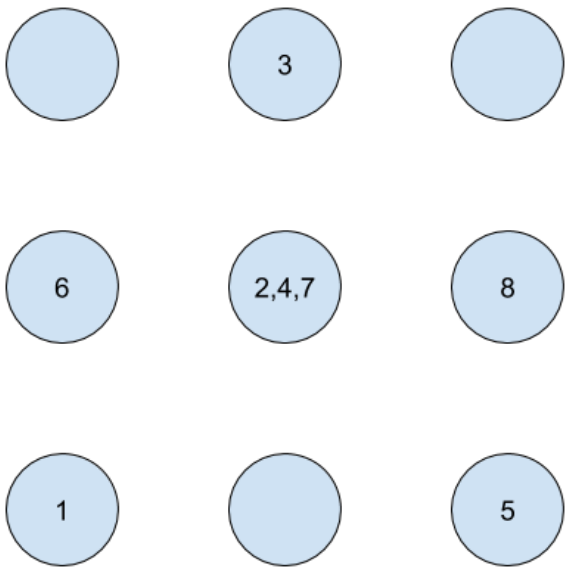


Figure 7. Letter “A” tactile trace from Yanagida et al (2004)

Although the identification of more complex patterns like those seen in Yanagida et al (2004) and Piateski and Jones (2005) have high rates of identification, localisation of specific tactor activation seems to be fairly poor. Sofia and Jones (2013) developed two 3 x 3 tactile arrays with an intermotor spacing of 22mm and 30mm. Users demonstrated a 49 percent localization accuracy of the forearm and a 46 percent localization accuracy on the thigh. **Design requirements 16.**

The materials used in the wearable arrays differ quite a bit. There’s loose fitting nylon suit with elastic straps (Gunther & O’Modhrain, 2002), stretch fleece (Van erp et al, 2002), stretch neoprene with velcro (Lindeman et al, 2004), spandex (Piateski and Jones, 2005), and a

"tight-fitting t-shirt" (Yang et al, 2002). The commonality between all of these materials is that they provide compression. **Design requirements 14 & 15.**

2.5 Haptics Exploiting the Funneling Illusion

Richter et al (2011) tested the funneling illusion using three different types of actuator; a coin type ERM vibrating motors, linear solenoids, and voice coils. The vibrating motors were attached to the ventral forearm using an "adaptable arm-sleeve". The solenoids and voice coils were embedded into a test jig which the user pressed their dorsal forearm onto. The inter stimulator distance was the same regardless of actuator and was set to 80mm. The users were able to perceive the funneling illusion with all three actuators but found that the vibrating motors were able to create the best results. Kato et al. (2010) induced the funneling illusion on the participants arm using two speakers with a 20mm diameter plastic contactor with a interstimulator distance of 130mm.

Cha et al (2008b) found that the optimal intertactor spacing on the forearm was 60mm. This contrasts with Alles (1970) who found that the optimal spacing between tactors was 127 mm on the forearm and 102 mm on the upper arm. In contrast, Kerdegari et al (2014) created a linear vibrotactile display using seven 10mm coin type ERM vibrating motors with a 25mm intermotor spacing. The motors are attached to a velcro strip which the user wears around their head. They utilized 1000ms duration stimuli. They utilized the one dimensional funneling illusion with various intermotor spacings (25mm, 50mm, 75mm, and 100mm) demonstrating that an intermotor distance of 25mm was far superior to the others.

The speed of the dynamic tactile stimulus seems to have an effect on the quality of the perception. Other research has demonstrated that a speed of 60 mm per second was optimal for users to perceive a continuous stimulus while speeds greater than 200 mm per second evoked the sensation of two distinct stimuli, and speeds less than 10 mm per second resulted in the perception of dimmer and disconnected stimulus (Cha et al., 2008b). For dynamic touch, a speed of 10-100 mm per second was found to be more pleasant to users than slower speeds of 3 mm per second or higher speeds of 300 mm per second (Ackerley et al., 2014). Pawling et al (2017) demonstrated that a stroking touch utilizing a soft brush at 30 mm/s were rated as more pleasant than 300 mm/s stimuli.

Rahal et al (2009a) used two coin type ERM vibrating motors attached to a strap of nylon which was wrapped around the users dorsal forearm. They tested the dynamic funneling illusion with two different rendering algorithms, one linear and one logarithmic, with a speed of 30mm/s with various intermotor distances (30mm, 40mm, 50mm, 60mm, and 70mm) for both a stimulus going longitudinally along the forearm or transversely across the forearm. The users' subjective preference was for the linear algorithm in nearly all permutations. They performed a second experiment (Rahal et al, 2009b) in which they used four coin type ERM vibrating motors attached to a strap of nylon which was wrapped around the users dorsal forearm. They chose an intermotor spacing of 50mm. They tested both the linear and logarithmic rendering algorithms and once again their users subjectively preferred the linear rendering algorithm.

One research team demonstrated an increase of the resolution of a vibrotactile display device utilizing the funneling illusion in one dimension (Barghout et al., 2009). Their vibrotactile

display device consisted of four coin type eccentric rotating mass (ERM) motors attached to the subjects forearm utilizing four pieces of velcro fabric which slightly compresses the motors into the forearm. The motors were placed 80mm apart to form a linear display with a length of 240mm. The display had 13 possible stimuli locations, four of which were the location of the motors. They utilized both a static and a dynamic funneling illusion on their display. The static stimuli had a detection rate of 31.5% and the dynamic stimuli had a detection rate of 32.69%

Mizukami and Sawada (2006) developed a novel shape-memory alloy device to serve as a small vibrotactile actuator for stimulating the palm and fingertips. They created a tactile array utilizing these devices in a 3 x 3 grid and running the actuators at 50Hz. This tactile array was placed on the palm of the user's hand and utilizing the one dimensional funneling illusion, three different characters were rendered (Figure 8). The authors claim that the letters were perceived by users but no experimental data was shown. These same actuators were used in another tactile display (Fukuyama et al, 2008). The tactile display was composed of 8 of these actuators in a 3x3 pattern (figure 9) driven at 50hz with an intermotor spacing of approximately 12mm. A left to right apparent motion was created by driving motors 1 and 3 for 400ms followed by driving motors 1, 3, 4, and 5 for 400ms followed by driving motors 4, 5, 6, and 8 for 400ms followed by driving motors 6 and 8 for 400ms. Following the same pattern, a right to left, a top to bottom, and a bottom to top apparent motion was created by the array. The 3 users that participated had a 90% success rate at determining the direction of the apparent motion.

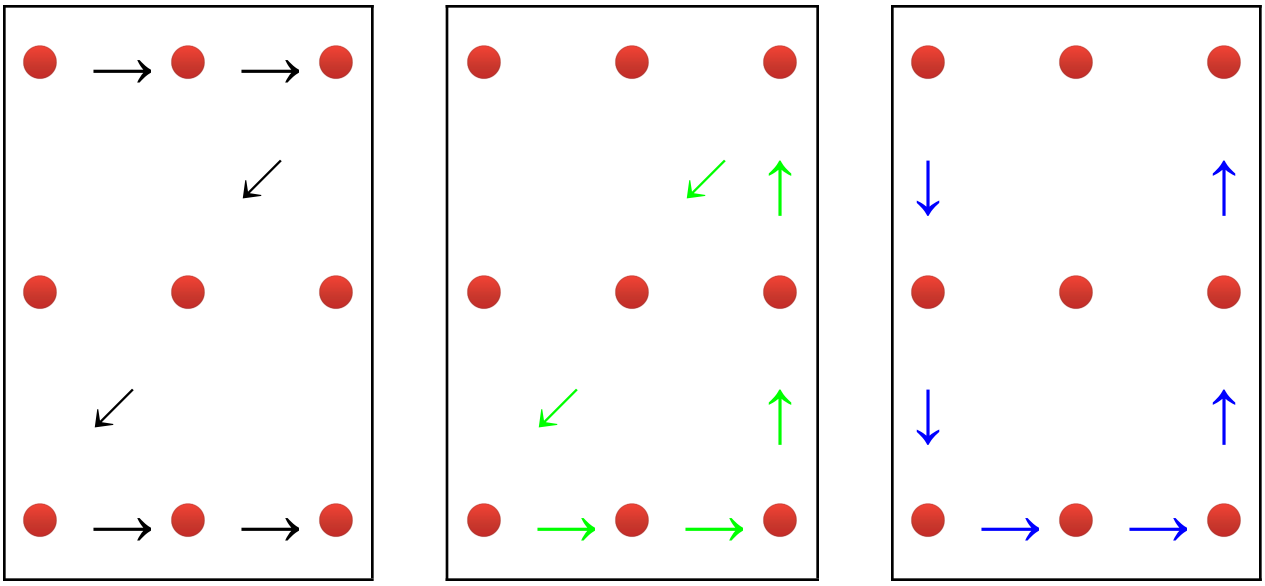


Figure 8. Red circles represent the actuators, the black arrows represent the movements for the "Z" character, the green arrows represent the movements for the "Δ" character, and the blue arrows represent the movements for the "U" character.

1	4	6
2		7
3	5	8

Figure 9. Motor Distribution Pattern

The mobile lorm glove (Gollner et al, 2012) was designed to allow the deaf-blind to communicate with other individuals which are unfamiliar with the Lorm alphabet. The Lorm is a hand touch alphabet in which the speaker can communicate with a deaf-blind individual by sequentially drawing the characters onto the reader's hand by tracing lines and shapes as seen in figure 10. The glove allows for two way communication, 35 fabric pressure sensors embedded into the palm of the hand allows for the deaf-blind individual to communicate using the Lorm

alphabet which is then sent via SMS to a recipient. There are 34 round, 10mm diameter pressure sensors which correlate with the characters of the Lorm alphabet and one larger rectangular pressure sensor to indicate the end of character input. Meanwhile, 32 8mm coin type vibrating motors located on the back of the hand allow for the recipient to send back an SMS which is translated into Lorm. Stroke based characters are performed by exploiting the one dimensional funneling illusion.

All this taken together seems to indicate that there is an optimal speed for the dynamic funneling illusion at around 60 mm/s, an optimal intertactor spacing of 25 to 120 mm dependant on body site, and that the optimal rendering algorithm will be linear.

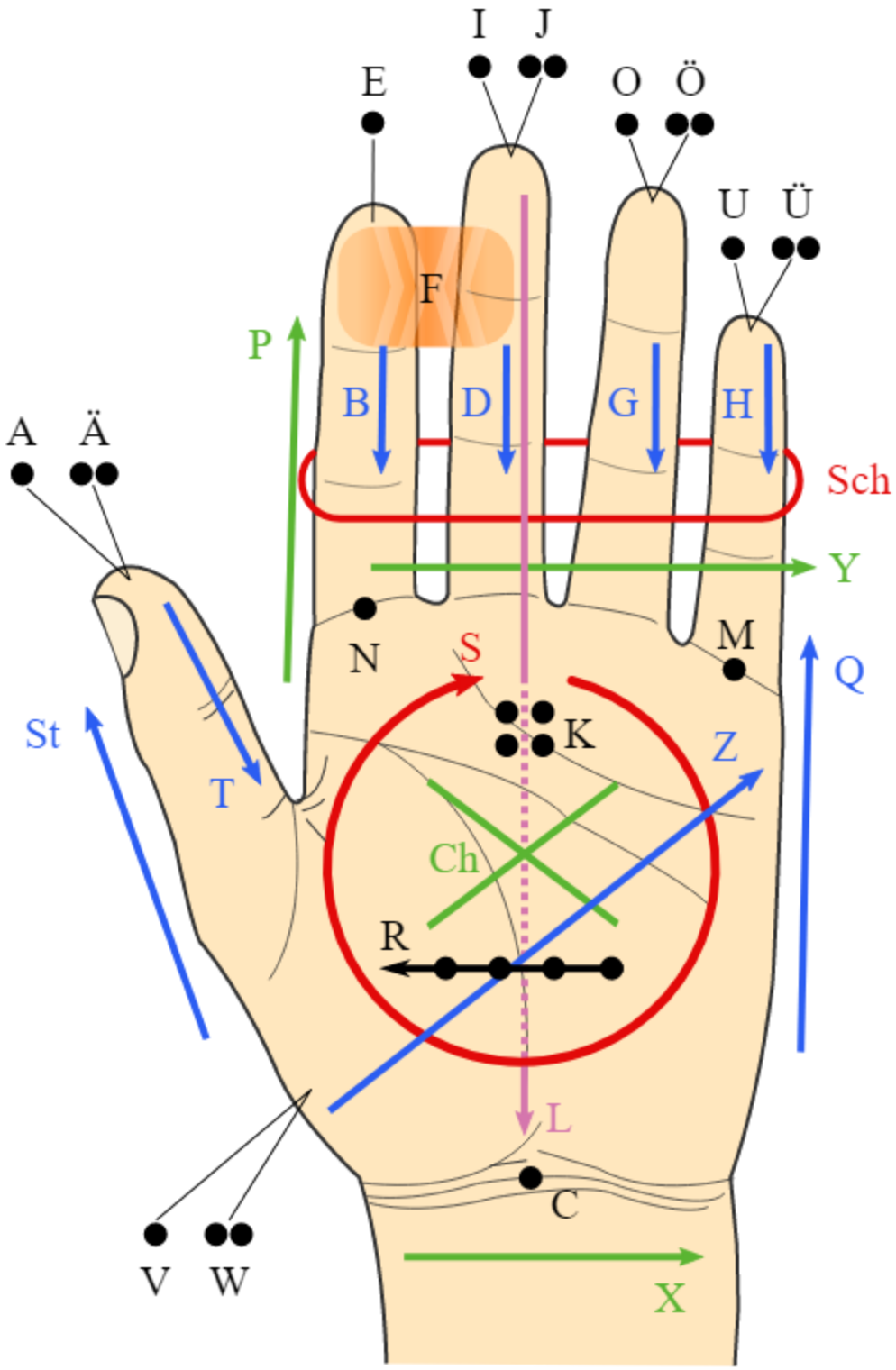


Figure 10. The German Lorm Alphabet. Flappiefh (2012), retrieved from URL (https://upload.wikimedia.org/wikipedia/commons/f/f7/Alphabet_de_Lorm_allemand.svg)

2.6 Design Guidelines

Previous research has been done to develop guidelines for the design of wearable tactile displays that we can draw from. We will first explore the work of Gemperle et al (1998) who developed 13 guidelines for wearability.

Their first guideline is about the placement of a wearable device and how the designer should take into consideration the dynamic nature of the human body. They identified three criteria that should be considered: first, use areas that are relatively the same size across adults; second, use areas that have low movement or flexibility even when the body is in motion, and third, use areas that are larger in surface area. The areas that meet these criteria are; the collar area, the rear of the upper arm, the forearm, the rear, side, and front ribcage, the waist and hips, the thigh, the shin, and the top of the foot.

Their second guideline is to have a humanistic form language for the wearable product with the goal of creating a safe, soft, and wearable form.

Their third guideline is to consider the dynamic structure of human movement including the mechanics of the joints, the shifting of flesh, and movement of muscles and tendons. They state that the two main ways of accomplishing this are by designing around the more active areas of the joints or by creating spaces within the wearable into which the body can move.

Their fourth guideline is to consider the human perception of space and that the wearable should not extend so far away from the body that it will interfere with the users ability to move through

space. They state that the wearable should stay within 0 to 127mm from the users body, dependent on which part of the body the wearable is protruding from.

Their fifth guideline is the consideration of sizing the wearable for body size diversity. They identify two ways of achieving this, first is the use of anthropometric data, and the second is the use of solid areas coupled with flexible areas.

Their sixth guideline is that comfortably attaching the wearable device to the user should be done by wrapping it around the body rather than using a fastening system like clips or shoulder straps.

For our purposes, their seventh guideline deals with being cognizant of the constraints that the containment of the embedded electronics brings to the design.

Their eighth guideline states that the weight of the device should be small enough not to hinder the users movement or balance and that the placement of weight should be distributed around the stomach, waist, and hip areas.

For our purposes, their ninth guideline deals with the wearable having proper access to the tactile modality of the users.

Their tenth guideline states that the sensory interaction with the wearable should be simple and intuitive.

Their eleventh guideline is to take into account the thermal properties of your device and how it will interact with the thermal properties of the user to ensure that heat is not focused or trapped.

Their twelfth guideline is to consider the aesthetics of the wearable and how culture, and context will influence the perception of the product.

Their thirteenth guideline is to test the long-term use of the wearable on the physical and mental well being of users. This is beyond the scope of this thesis.

Next, Gemperle et al (2001) created a list of guidelines for the design of a wearable tactile array.

The first guideline is that the tactile array should be light weight which is a specific rehashing of the eighth guideline that was previously discussed. The second guideline is that the tactile array should be silent. The third guideline is that the tactile array should be low power. The fourth guideline is that the tactile array should be discreet. The fifth guideline is that tactors should be small. The sixth guideline is that the tactors should be felt through clothing. The seventh guideline is that tactors must be held tight to the body. The eighth guideline is that the tactile array should support flexible tactile information design.

Jones & Sarter (2008) gave some guidelines for the design of tactile displays. They state that these displays should be lightweight and small. They should minimize power consumption. Tactors should be durable, cost effective, reliable, wearable, power efficient, robust, and have a wide dynamic range. The tactile display should maintain continuous contact with the user when the user moves dynamically. The tactile signals should be of sufficient magnitude that they are attended to but not aversive.

A previous attempt at a scalable system for controlling tactile arrays was undertaken by Lindeman and Cutler (2003). They identified several factors that their system should meet:

- Small unit size
- Light weight
- Low power consumption
- Ease of putting on and taking off
- Low cost
- Integratable into existing systems
- Easily configurable
- User Extensible
- Battery powered and portable
- The number of components should be restricted to reduce the complexity of assembly and troubleshooting
- It should be possible to update the software on the tactaboard in the field.
- Support 16 outputs per board and to support multiple boards on a single communication line.

They originally wanted to allow for 256 discrete analog output levels but were unable to figure out a way to do this efficiently. They settled on using pulse width modulation (PWM) instead and this also had the advantage of being more power efficient than their analog approach.

They named the system they designed the Tactaboard. The PCB measures 89mm x 114mm and it's mounted in a 110mm x 190mm x 58mm box. The tactors are connected to the box with a 2.5mm headphone jack. The box has two power supplies, one for the board and one for the tactors. They utilized a 3.5mm stereo headphone jack as a serial connector. They have a serial interface with a custom control protocol with a communication speed with the host computer ranging from 2400 bps to 115200 bps. They developed an API that allows for the setting of 16 PWM outputs to 256 discrete levels. For their tactors, they utilized coin type 18mm vibrating motors. Figure 11 shows a tactile array using the TactaBoard that Yanagida et al (2004) developed using the same tactors.

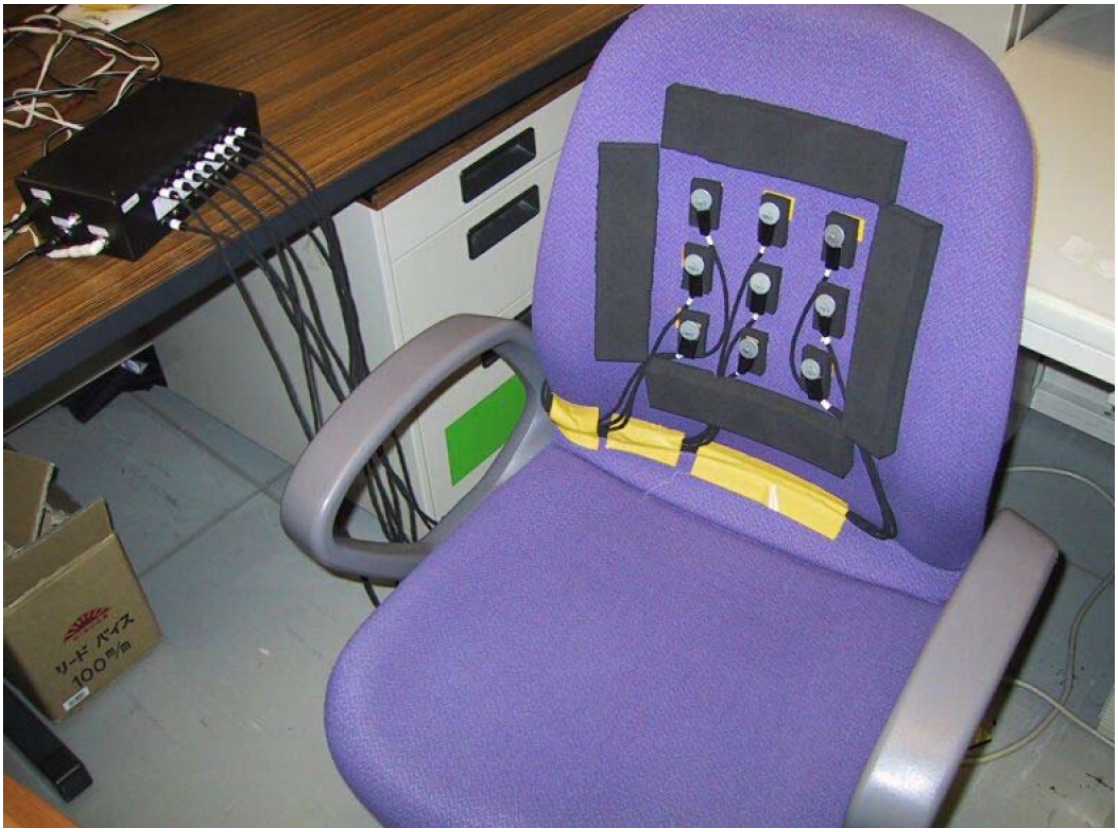


Figure 11. Tactile Array using the TactaBoard from Yanagida et al (2004). © 2004 IEEE.

Their system failed to meet several of their factors. Their unit size was too large, which made it non wearable, it was not battery powered, and it did not support multiple boards.

3.0 Design

Now that we have seen what others have created and have the needed parameters to effectively render the funneling illusion, we can set out to design our wearable tactile array. First, we will come up with a set of design requirements that the final wearable tactile array should meet. The design process for achieving these requirements for the soft goods and electronics is described in section 3.1 and the software design process is described in section 3.2.

3.1 Wearable Tactile Array

We can use the design guidelines that we have discussed along with the previous attempts to create tactile arrays covered in the haptics section to make our requirements for the wearable tactile array that we will design. There are 20 requirements that should be met.

1. The placement of the wearable tactile array should be on a large surface area with limited movement and flexibility.

2. The wearable tactile array should feel safe and comfortable on the user.

3. The wearable tactile array should not protrude away from the users body such that it interferes with their natural movements and ability to move through the environment.

4. The wearable tactile array should fit various body sizes.

5. The wearable tactile array should be light enough that it does not encumber the user.

6. The tactile signals should be of sufficient magnitude that they are attended but not aversive.

7. The wearable tactile array should be aesthetically pleasing.

8. The wearable tactile array should be relatively silent.

9. The wearable tactile array should have low power consumption so it can be untethered if necessary.

10. The wearable tactile array should be easy to put on and take off.

11. The components of the wearable tactile array should be cost effective.

12. The tactors should be small.

13. The intertactor spacing should stay consistent even when the user moves.

14. The tactors should be held tight to the body.

15. The tactors should maintain contact with the user when the user moves.

16. The vibration from each tactor should be isolated to that tactor.

17. The system should be integratable into existing systems.

18. The system should be easily configurable.

19. The system should be expandable.

20. The delay in the system should be less than 25 ms.

The torso is the body surface which best fits requirement 1. This is therefore our target for the design of the wearable tactile array.

In order to meet design requirements 11 and 12, small coin type eccentric rotating mass (ERM) motors were chosen as the tactors. The propagation of vibration along the skin drastically attenuates after 8mm from these types of tactors (Sofia & Jones, 2013) which will help with requirement 16. The chosen tactor was the NFP-C1034 which is a 10mm diameter motor rated at 3.0V with a frequency of 203.3Hz and a vibe force of 1.35G. According to Verrillo (1966), the maximum sensitivity of hairy skin to vibrotactile stimulation is at 220Hz which should make these motors optimal although Sofia and Jones (2013) demonstrated that coin type ERM vibrating motors lose about 37% of their frequency when mounted on the hairy skin which would put the frequency at around 128Hz. The current draw was tested on the motors. Each motor was attached to a power supply and taped to the benchtop with a taut piece of electrical tape. The average power consumption at 3 volts was 91.84 mA and the standard deviation was 3.41 mA. The raw data can be found in appendix A and a histogram can be seen in figure 12.

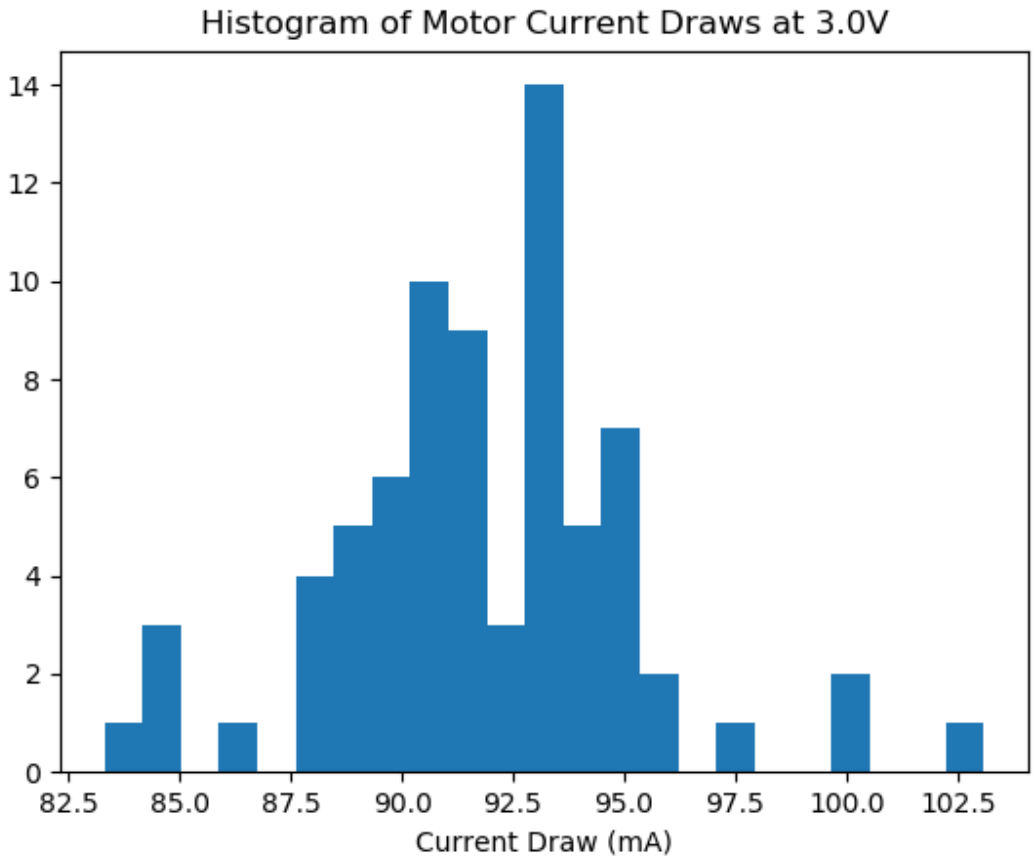


Figure 12.

Subjects found tangential stimulation to impart a better defined sensation than perpendicular stimulation and that this may be due to perpendicular stimulation transmitting more vibration through the skin stimulating more nerves and producing a less distinctive sensation (Alles, 1970). This indicates that our tactors should be mounted to impart tangential stimulation.

The design of the system started on the small scale with a 2 x 2 array of tactors in order to test out different materials.

The first material that was tested was a canvas type material with the tactors glued on (figure 13) with some tensor bandages sewn to the edges in order to be able to keep proper contact with the users skin (figure 14). This was not particularly comfortable for the user and it was not durable.

Figure 13. Early cloth prototype with tensor bandages

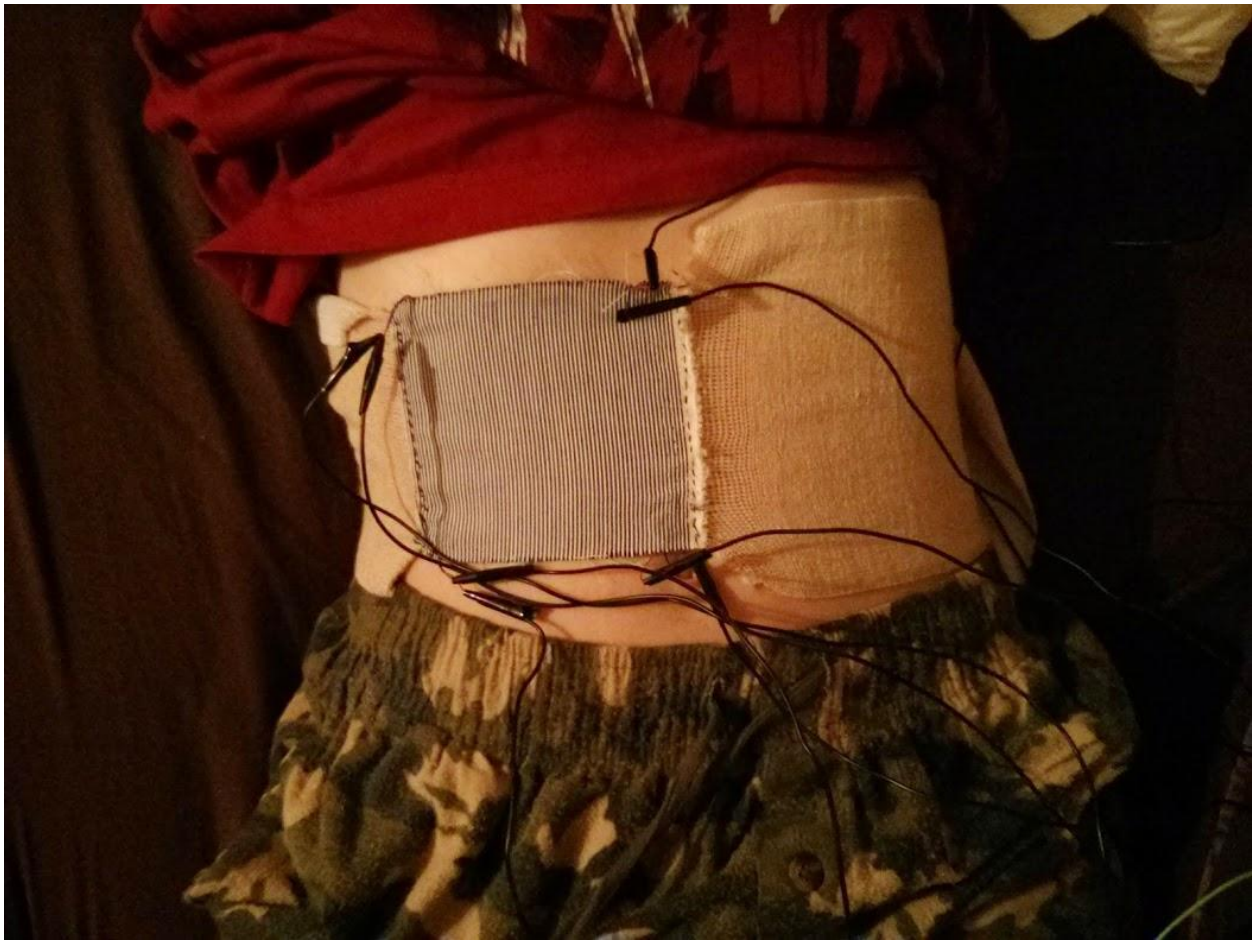


Figure 14. Early cloth prototype worn around the waist.

The second material that was tested was a modular 3D printed plastic system. The thought was that using a ball and socket design would allow for the needed flexibility to maintain contact as the user moves while keeping the intertactor distance consistent (figure 15). The 3D printed material was both too fragile as it was easily broken and too stiff as it failed to isolate the vibrations from each tactor.

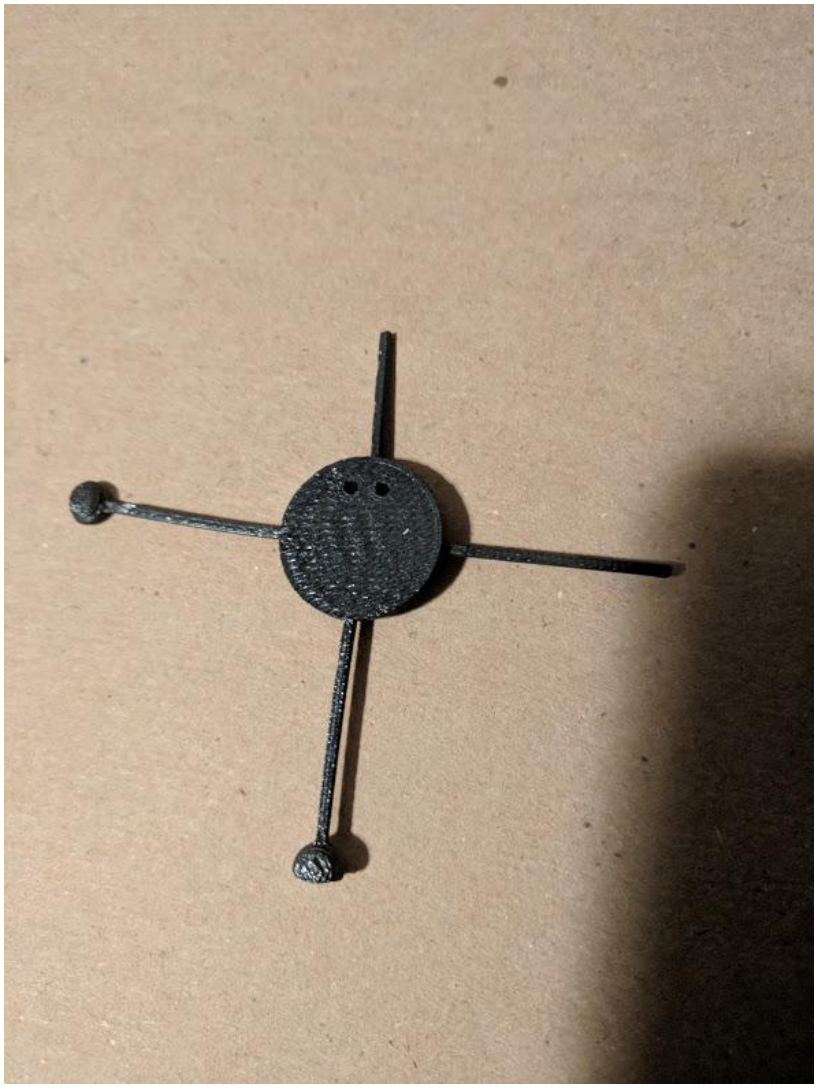


Figure 15. 3D printed modular plastic prototype.

In order to maintain the intertactor spacing while isolating vibrations, the next material that was tested was a 3D printed digital rubber. The first prototype (figure 16) was too thin and it was unable to maintain the intertactor spacing and it also was weak and broke. A broader area of rubber was used for the second prototype (figure 17), this time it was able to maintain the intertactor spacing but was unable to adequately isolate the vibrations from each tactor.

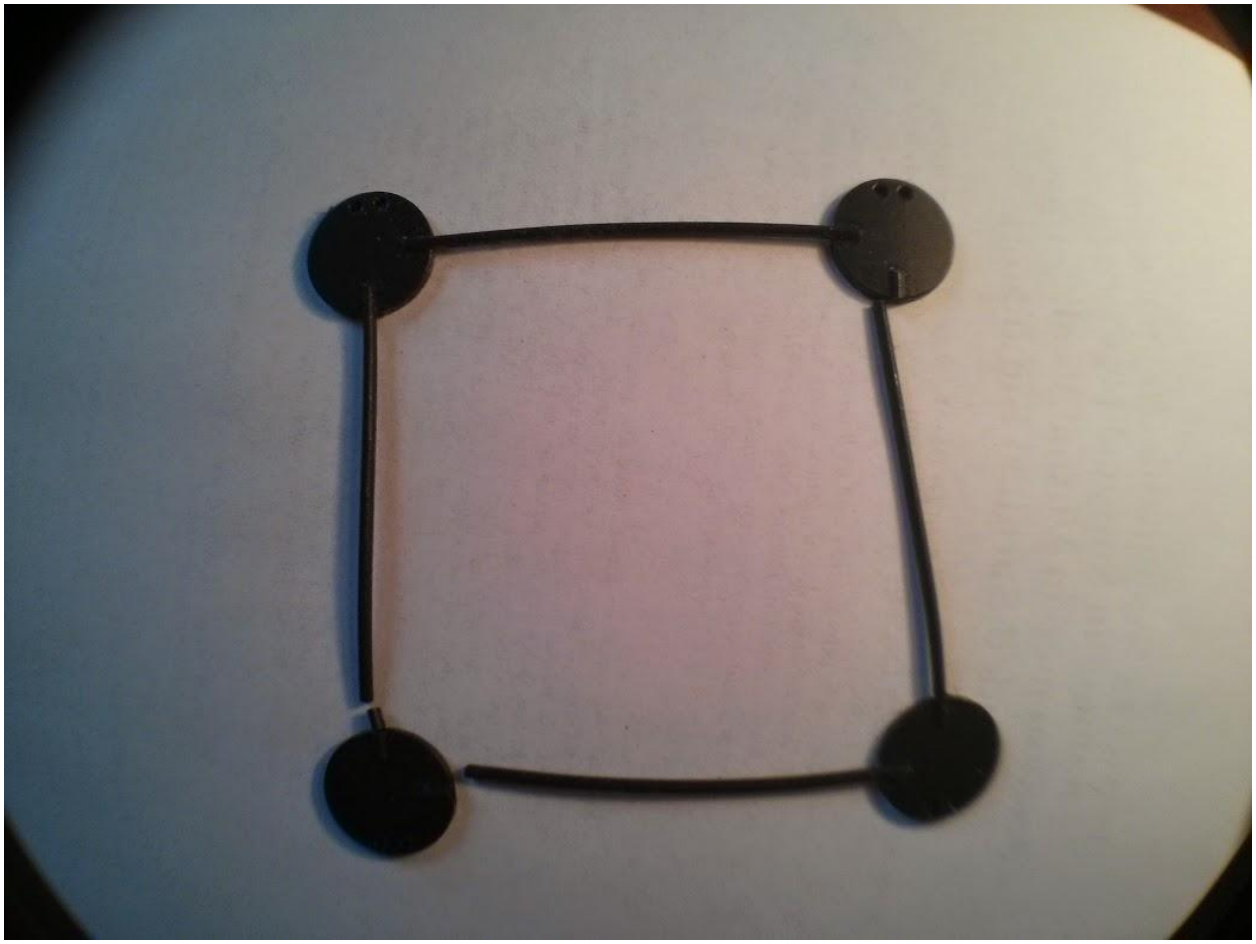


Figure 16. 3D printed digital rubber prototype 1.

Figure 17. 3D printed digital rubber prototype 2.

The next material that was tested was a 10mm thick very low density open cell foam (figure 18). This material isolated the vibrations extremely well while being stiff enough to maintain the necessary intertactor distance. The material is also very light and flexible enough to move with the range of motion of the user and the open cell nature of the foam makes it so that it does not trap heat. This material meets all the necessary requirements. A foam panel with a 4 x 6 array of tactors to be used on the torso can be seen in figure 19.

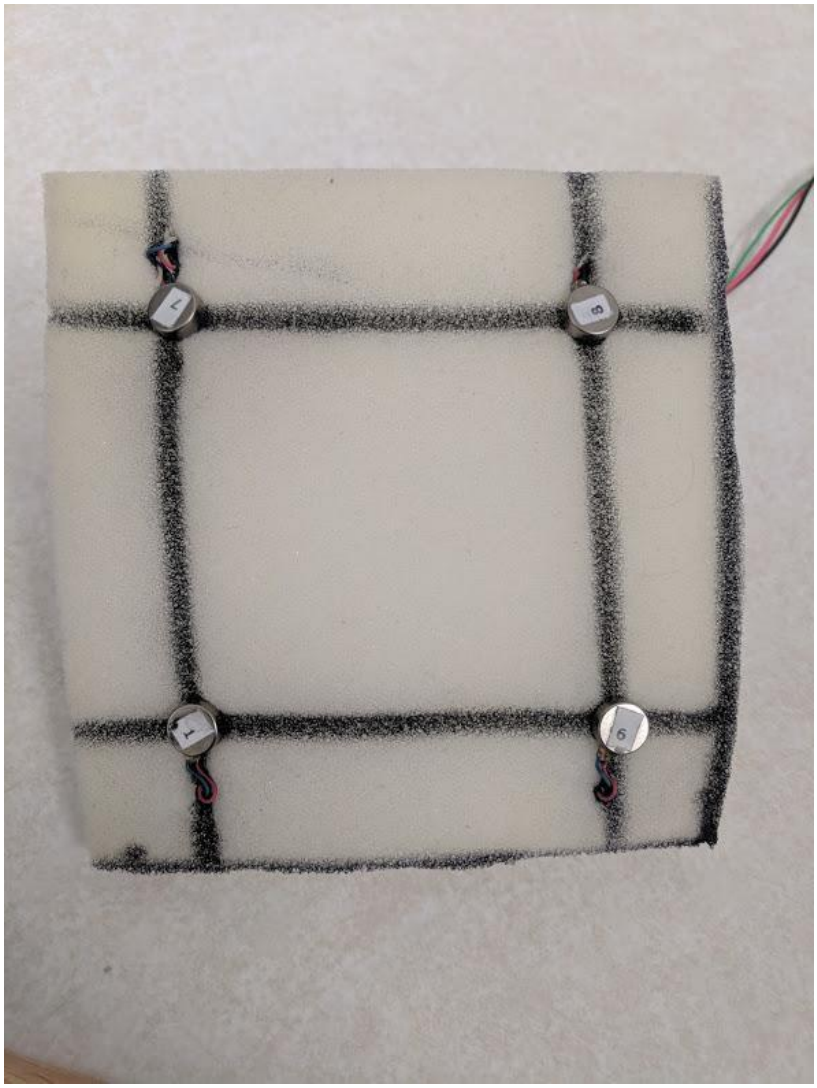


Figure 18. Low density open cell foam prototype.

Figure 19. Foam panel for wearable tactile array.

The next design challenge was to make the foam panel wearable. This was accomplished with the use of a compression material. The material is a blend of 80% polyester and 20% elastomer which compresses the panel onto the user keeping it in place. The material is very soft and has a nice feel against the skin while being very breathable and it has moisture wicking properties. The compression material was sewn into a vest form factor with a pocket in the front and a pocket in the back to be able to fit two panels into the system (figure 20).

Figure 20. Garment for wearable tactile array.

The next step is to be able to effectively control the tactors. The first thought with most electronics projects is to use an Arduino board. I used the Arduino Mega which worked for controlling 4 tactors but it did not give the necessary flexibility needed for scaling up the system.

The next controller to be used was a Raspberry Pi 2 Model B used in conjunction with custom breadboard circuitry. Figure 21 shows the breadboard circuitry for controlling 4 motors. The system was scaled up to a 4 x 6 array which can be seen in Figure 22. The number and length of

wires needed for this system as well as the size of the breadboard circuitry for controlling 24 motors made this design unfeasible, although it did demonstrate that the algorithm developed did work across the entire grid.

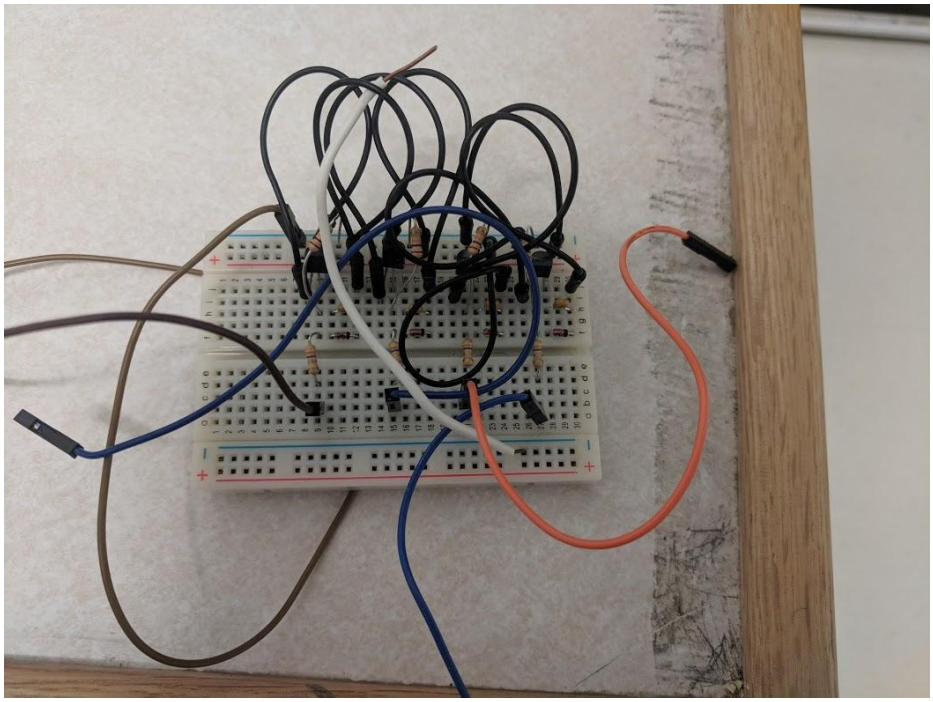


Figure 21. Breadboard circuit

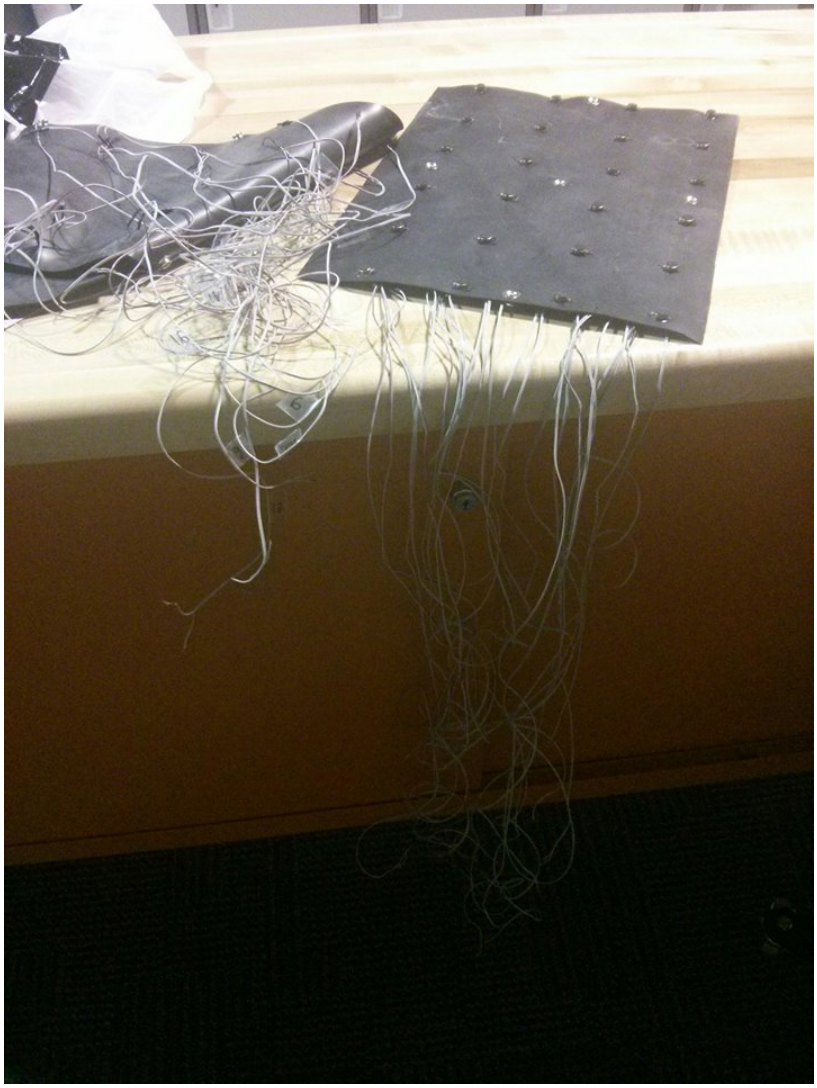


Figure 22. Rubber panel prototype.

If connecting every tactor in the array to a single board then a modular system needed to be designed. An electronics engineer created custom PCBs with each PCB being able to control the intensity of one motor and being able to pass information along to the following PCB in the daisy chain. The custom PCBs and their enclosures can be seen in Figure 23. This system worked very well and was successfully miniaturized (Figure 24). There was one downside to the system; the more boards you daisy chained together to create a larger array, the larger the amount

of information needing to be passed on was and the more communication steps there were. For example, with one byte of information per board with four boards, four bytes of information needs to be passed to the first board, three to the second board, two to the third board, then one to the fourth board. With a scaled up system this can introduce a perceptible delay.

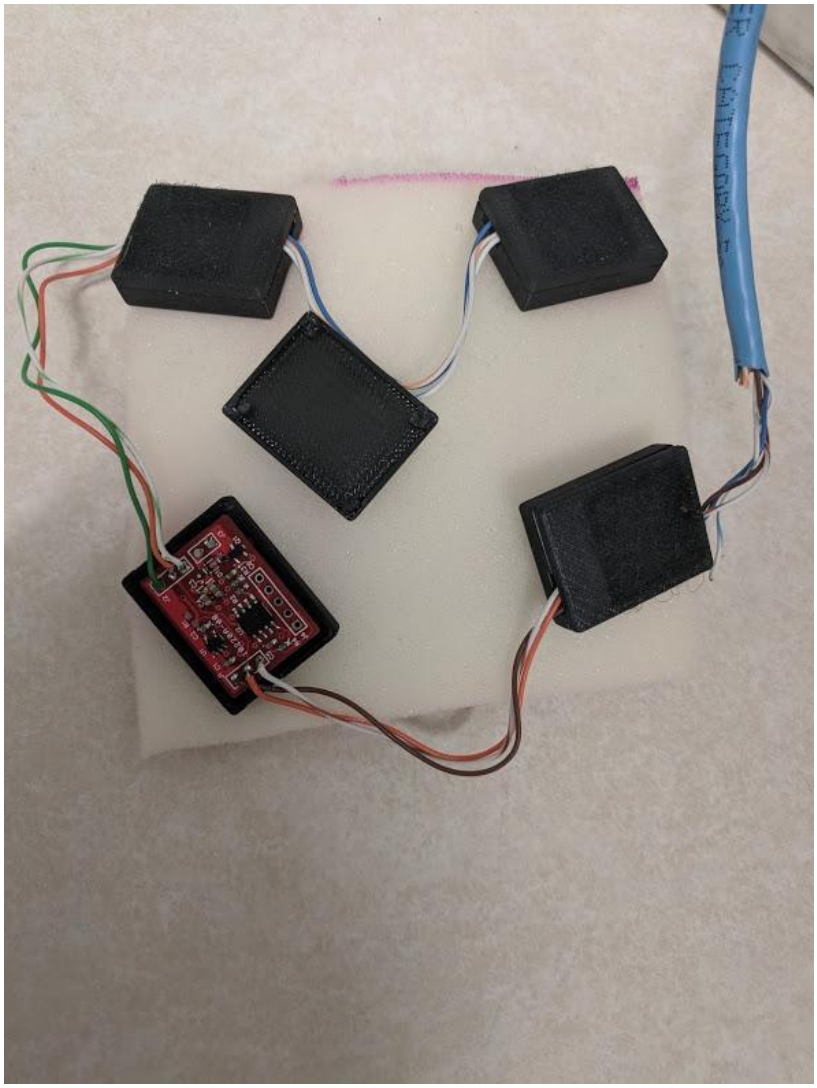


Figure 23. Modular PCBs version 1 with 3D printed plastic housings.

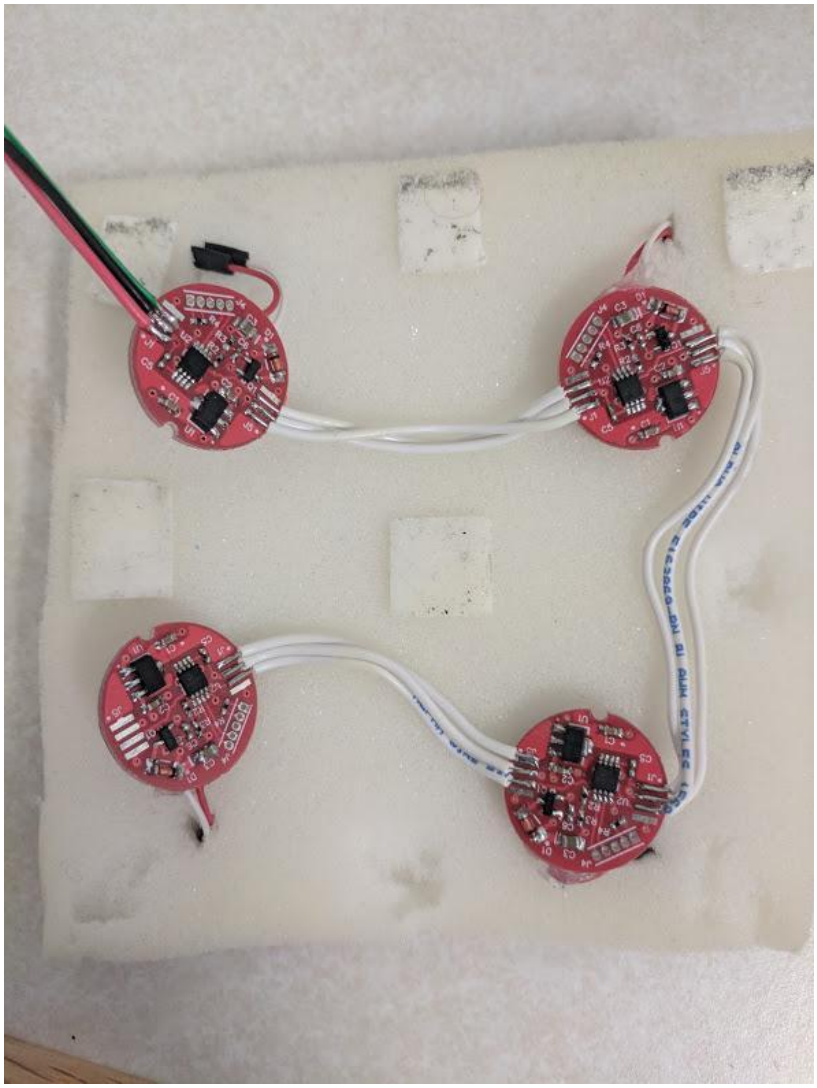


Figure 24. Modular PCBs version 2.

To cut down on this delay, a modular PCB was designed that was capable of controlling four tactors (Figure 25). This allows for the control of a 4 x 6 tactile array with the use of 6 boards (Figure 26). While attached to the panel, the current draw of each of the 6 boards was tested at 5 different PWM duty cycles (0%, 24.6%, 50%, 75.4%, and 100%). The raw data can be found in appendix A and a bar chart can be seen in figure 27. With a mean current draw of 7.3 mA and 379.9 mA per board at a 0% and 100% duty cycle respectively, we can say that requirement 9

has been achieved. As seen in figure 26, the loose connections between the modular PCBs allows for flexibility of movement of the user. The size of the PCB enclosures is 45mm x 45mm x 13mm.

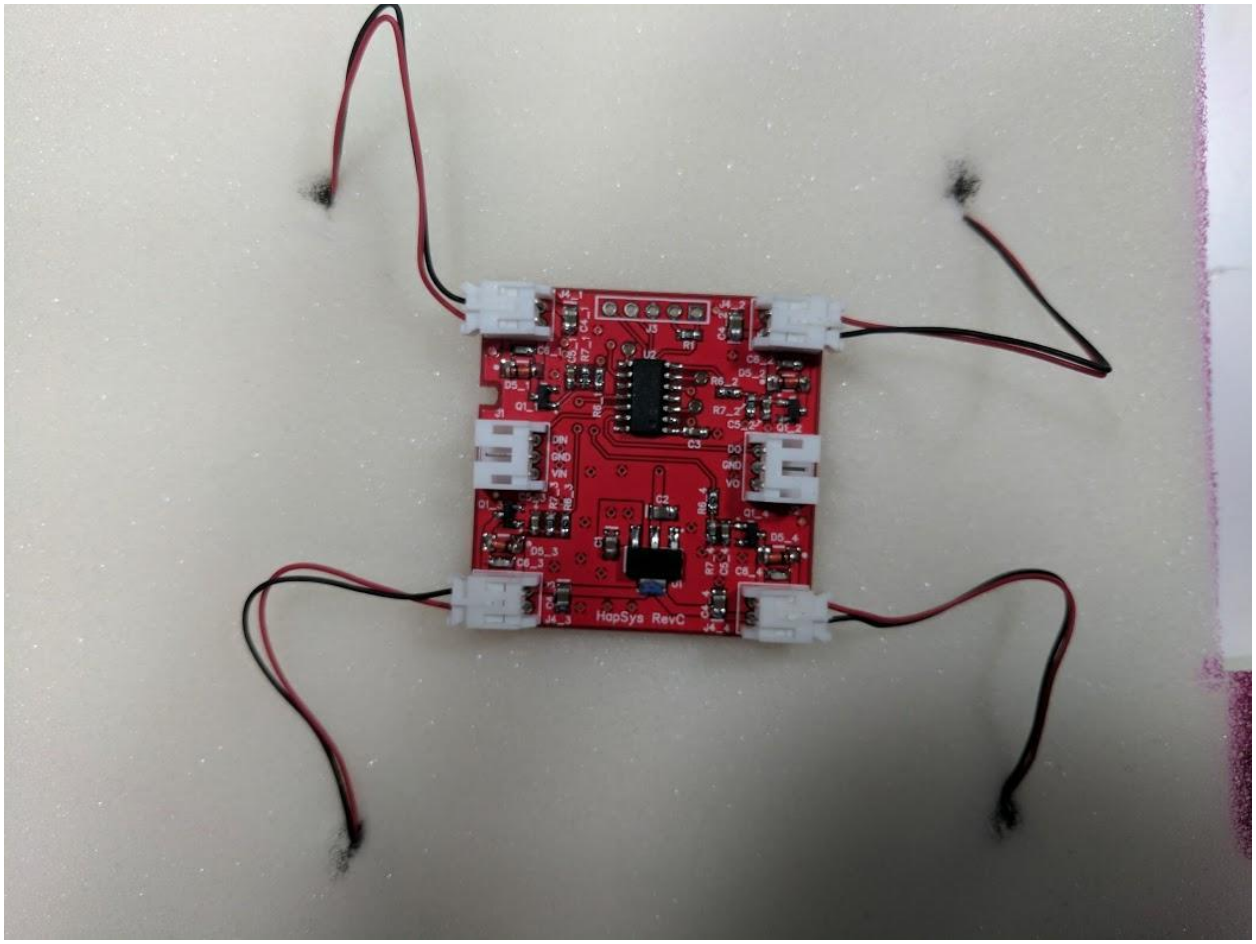


Figure 25. Modular PCB version 3.

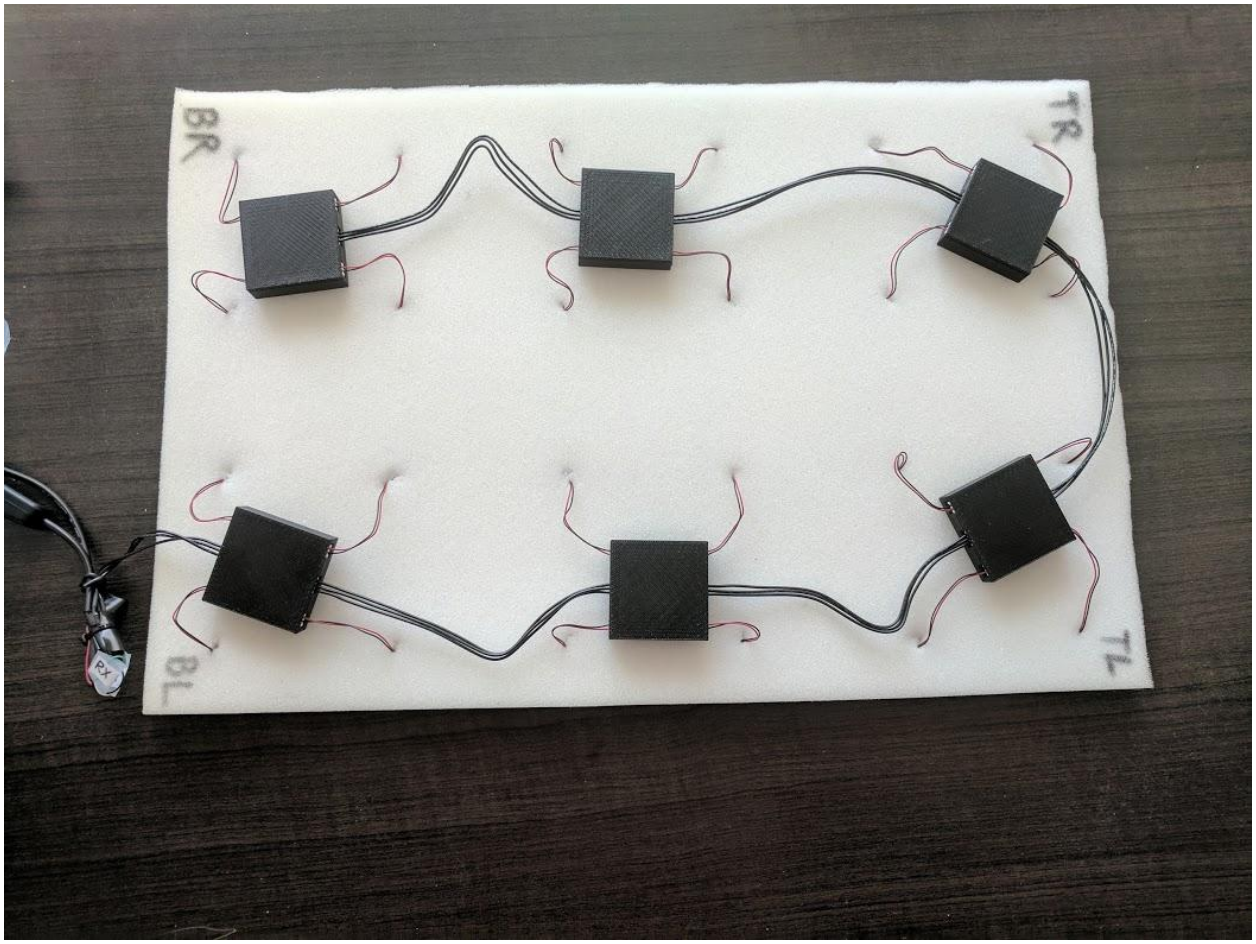


Figure 26. Back side of foam panel with attached modular PCBs in plastic housings.

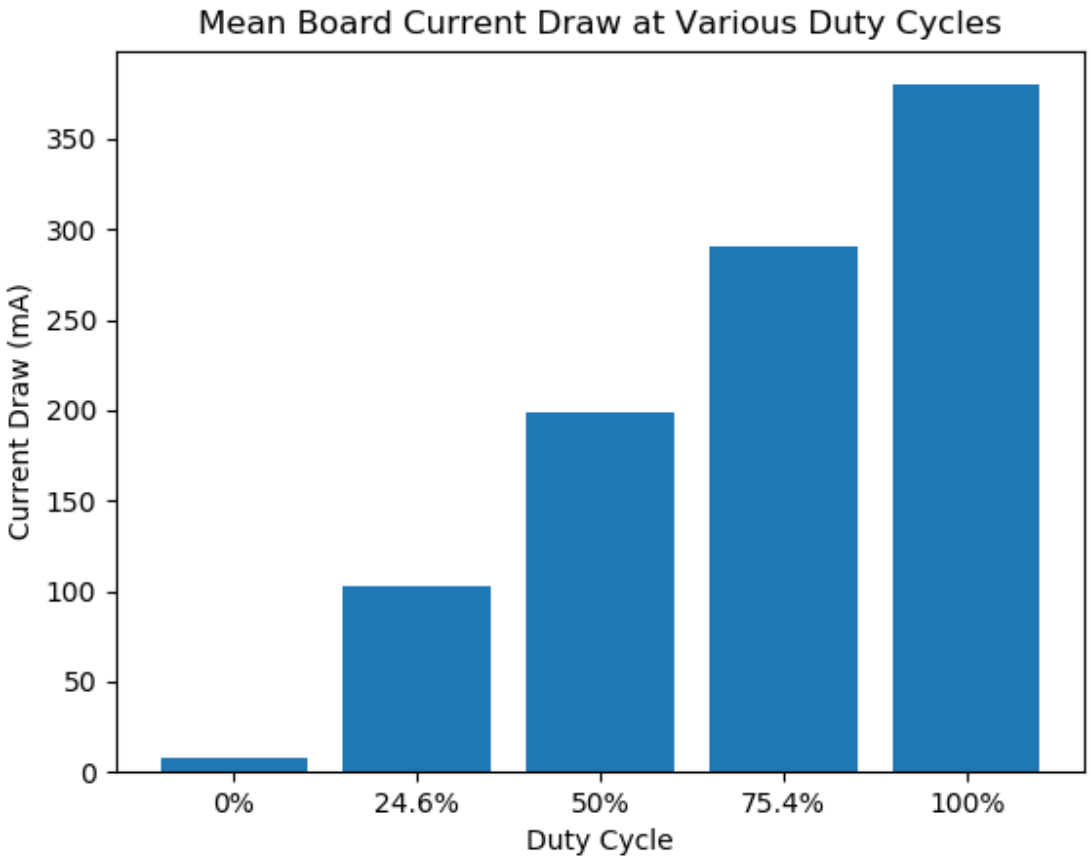


Figure 27.

In order to facilitate connecting the vest to the computer and to reduce the delay in a two panel configuration, an interface PCB was designed (Figure 28). It has a micro USB connection for data from the computer and a DC power jack for external power. The PCB allows for the data stream from the computer to be segregated into one stream for the front panel and one stream for the back panel which can then be propagated in parallel. The size of the interface pcb enclosure is 65mm x 60mm x 15mm.

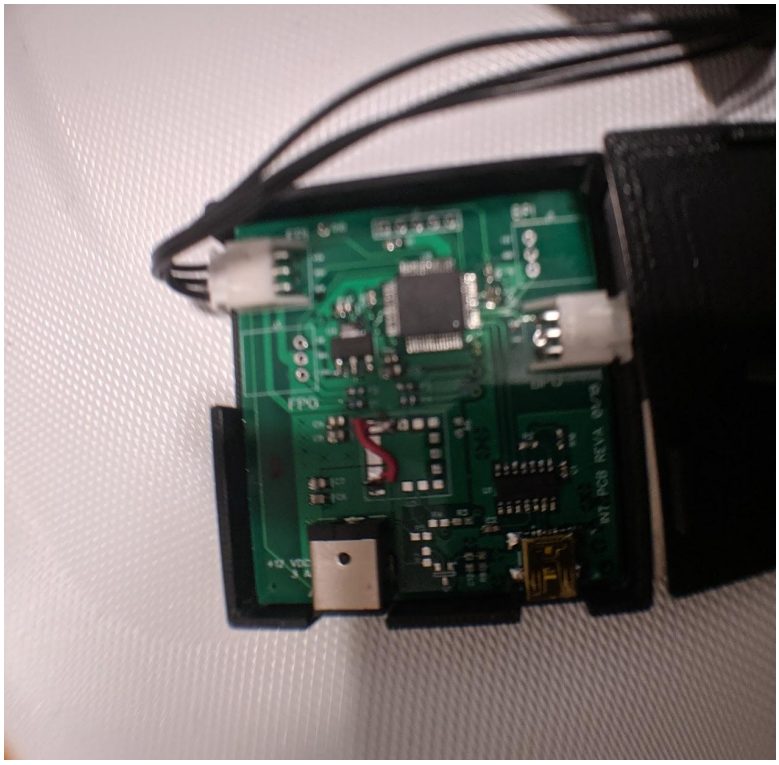


Figure 28. Interface PCB.

The final wearable tactile array design can be seen in figure 29. It is composed of 48 motors, 24 on the front and 24 on the back to give full coverage of the torso for tactile feedback. Two foam panels giving enough structure for the motors to stay where they need while isolating the vibrations. Twelve modular motor control PCBs along with an interface PCB allow for the control of the tactors within milliseconds. And to tie it all together a compression garment that helps keep everything in place while allowing for great contact between the motors and the users skin. The wearable tactile array weighs 694g.

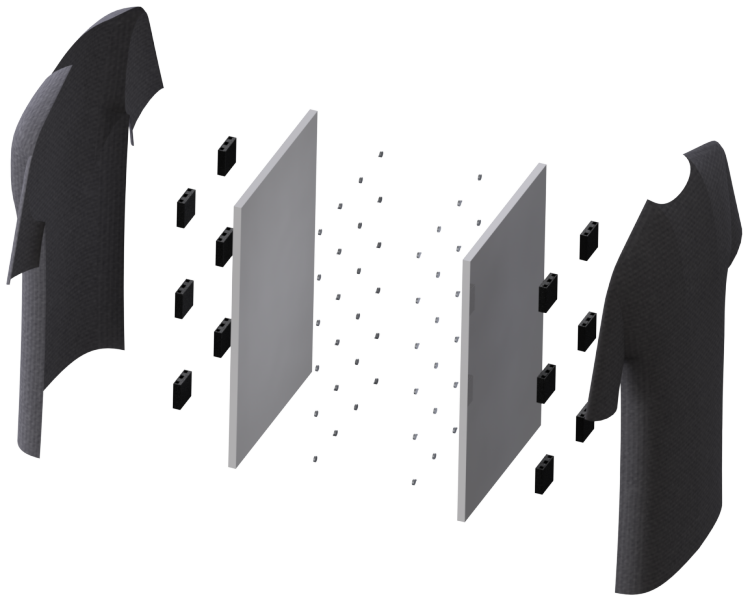


Figure 29. Blow up render of the wearable tactile array.

With the wearable tactile array design, all but one design requirement was reached. Because of the constraints on intertactor spacing, it made it impossible to meet design requirement 4. Because of the flat panel design that was needed, it may cause the wearable tactile array to not maintain proper contact over the breast area in female users.

3.2 Software

Now that we have finished the discussion of our hardware design process, we will now begin our discussion of the software design process. The focus of the software design is to be able to achieve requirements 17, 18, 19, and 20. Someone with limited programming experience and no technical knowledge of tactile perceptual illusions should be able to control the wearable tactile array. To achieve this a software development kit (SDK) was developed.

The SDK utilizes a cartesian plane metaphor to facilitate the understanding of what can be done. The front and back panels can each be seen as a cartesian plane with the bottom left tactor being coordinates (0, 0) and the top right tactor being coordinates (767, 1279), this can be seen in Figure 30.

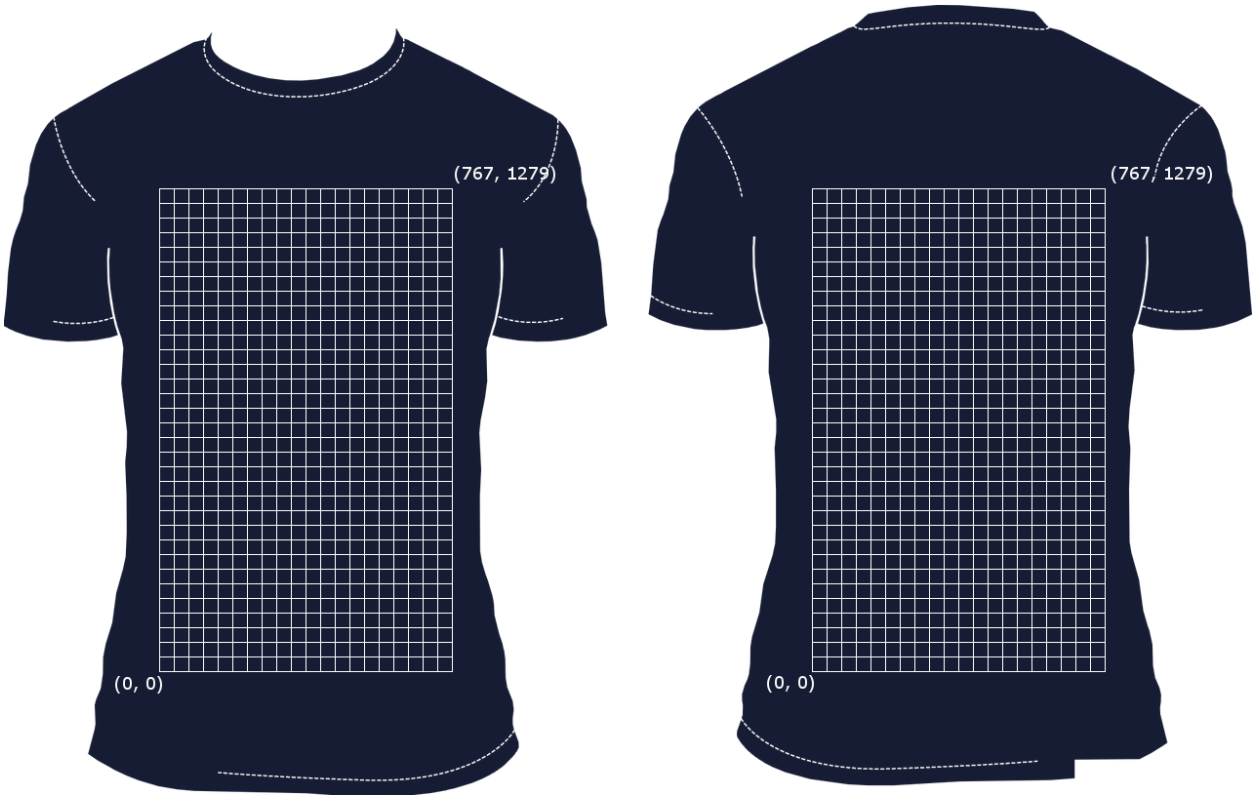


Figure 30. The cartesian planes representing the front (left) and back (right) panels.

Any static point on the cartesian plane can be rendered tactilely by exploiting the funneling illusion in two dimensions. This data structure is referred to as a tactile point. Any dynamic stimulus can be rendered by simply creating a series of tactile points with proper interstimulus

intervals. This data structure is referred to as a tactile object. Both the tactile point and tactile object are rendered by a novel algorithm

As we have seen in previous sections, the funneling illusion can be induced by utilizing two tactors to stimulate the underlying receptors when certain conditions are met (proper intertactor distance, proper stimulus duration, ect…). This illusion seems to be a property of higher order receptive fields. These receptive fields are two dimensional, therefore the funneling illusion should work in two dimensions.

As we have seen, the perceived stimulus is funneled towards the tactor with the most intensity. We will create a linear gradient of possible intensities between 0 and the maximum intensity, with the maximum intensity occurring when the perceived stimulus point (tactile point) is directly on the tactor and 0 intensity occurring when the perceived stimulus point is the inter tactor distance away. Let ItD be the inter-tactor distance, D be the distance between the tactor and the perceived stimulus, and MI be the maximum intensity. The intensity of the tactor will be:

MI - (D*MI / ItD)

where the maximum value of D is ItD. We will then calculate the intensity for all the motors which make up the array the perceived stimulus is in. We now have the ability to create a perceived stimulus at any point we wish within the array, we now need to give it movement. This brings us to the second piece of information; the delay between subsequent perceived stimulus points needed to propagate the stimulus at the chosen speed. We'll let IsD be the inter-stimulus distance and S be the chosen speed. The needed delay will simply be:

IsD / S

In order to render a tactile object, such as a circle, we simply need to calculate all the necessary perceived stimulus points needed within the array and the delay needed between the creation of each perceived stimulus point as outlined above.

The information to render each tactile point is then sent to the wearable tactile array as a 48 byte binary array at a speed of 460800 baud. The data flow through the system can be seen in figure 31. From the data flow diagram we can see that the maximum amount of time for data to be transmitted through the system will be 156 bytes (48 + 48 + 20 + 16 + 12 + 8 + 4). With each byte being a data frame of 10 bits, this means there are 1560 transmitted bits. 1560 transmitted bits at a rate of 460800 baud equals approximately 3.39 ms. This puts us well within the maximum 25 ms delay of requirement 20.

The choice of writing the SDK in C# means that the wearable tactile array should be easily integratable into existing systems (requirement 17). The SDK is easily configurable into new system designs by simply changing the node maps and the error checking function (requirements 18 and 19). The full documentation for the SDK can be found in Appendix 2 and the SDK can be found on GitHub (https://github.com/kcayen/ThesisSDK).

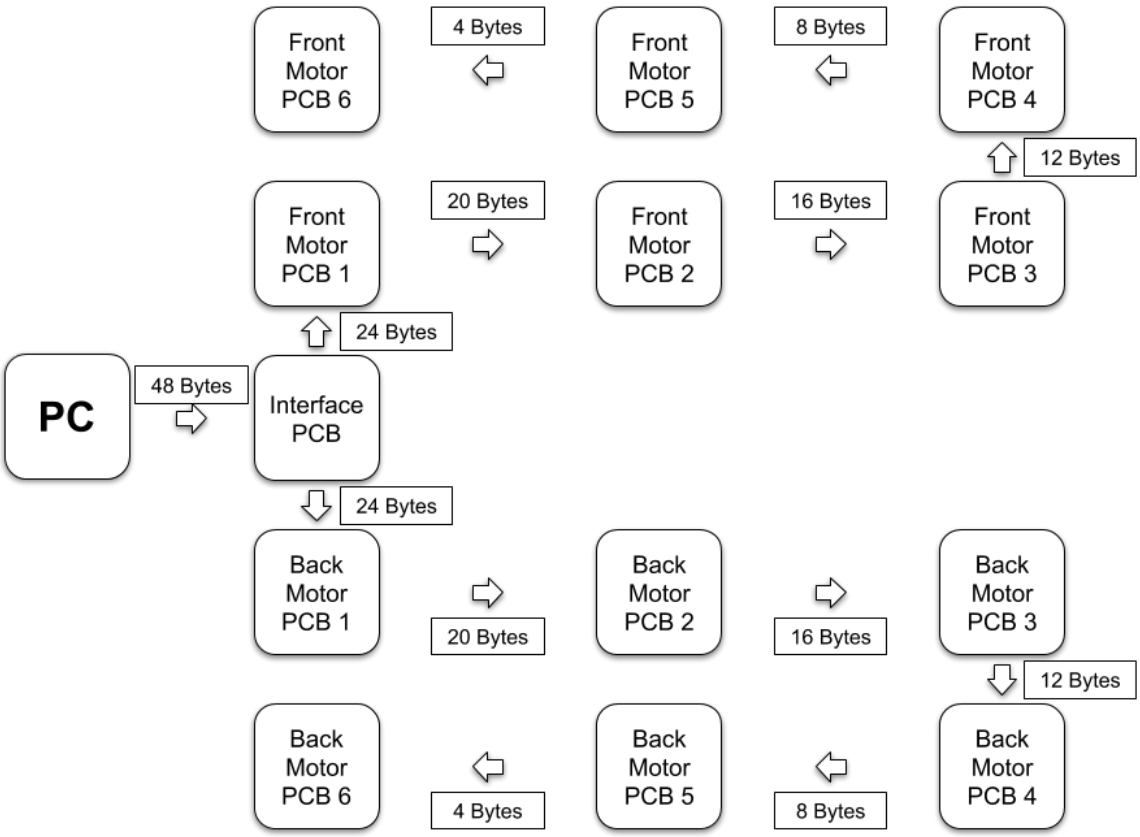


Figure 31. Data flow diagram.

4.0 Experiment Methodology

An experiment on tactile shape discrimination was done to test the novel two dimensional funneling illusion rendering algorithm. We will look at the method used to test in algorithm (section 4.1) and go through the step by step procedure of the experiment (section 4.2). The hypotheses for this experiment are:

H_0: The participants will not be capable of discriminating the rendered circle stimulus from the rendered square stimulus and thus their accuracy for discrimination will not differ from 50%.

H_1: The participants will be able to discriminate the rendered circle stimuli from the rendered square stimuli at above chance levels.

4.1 Method

The participants were outfitted with the wearable tactile array as well as headphones playing white noise (Relaxing White Noise, 2015) and were tasked with determining whether a circle or a square was rendered on the front panel of the vest. The size and location of the stimuli were chosen so that the majority of the rendered tactile points could only be done with a two dimensional funneling illusion algorithm (Figure 32). The size of the circle and square stimuli are matched to make the differentiating factor the curve of the circle.

A choice had to be made to either match the rendering speed of the dynamic illusion or to match the total rendering time of the shape. The thought was that the difference in rendering speed (120

mm/s vs 153 mm/s) was going to be less perceivable than the difference in total rendering time with matching speed (4.2 seconds vs 5.35s) and therefore the choice was made to match the total rendering time.

Another thing to consider was the starting point for each tactile shape. Although there aren't any overlapping tactile points between the two tactile shapes, there are some that are close enough that it would be difficult to pinpoint the difference in starting point. Also, with the same starting point, the participants could start to use a strategy of only focusing on one segment of the stimulus to differentiate the two. Therefore, 8 different starting points were utilized for each tactile shape (the cyan and grey points on Figure 32).

The experiment was designed in the Unity3D game engine along with the developed SDK. The Unity project can be found on GitHub (https://github.com/kcayen/Thesis_Experiment). The participants were presented with a total of 160 rendered stimuli (2 shapes x 8 starting points x 10 each) in a random order. After each rendered stimuli they were forced to choose between two options; did they perceive a circle or did they perceive a square. After the experiment, the participants filled in a short questionnaire which can be found in Appendix C.

A total of 11 male participants took part in the experiment. Eight of the participants were between the ages of 18 and 24, two of the participants were between the ages of 24 and 29, and one participant was between the ages of 40 and 49. Ten of the participants were from the computer science department. Only male participants were used to ensure that the wearable tactile array fit properly on the participant. All procedures were approved by the Laurentian University Research Ethics Boards (#2013-07-15).

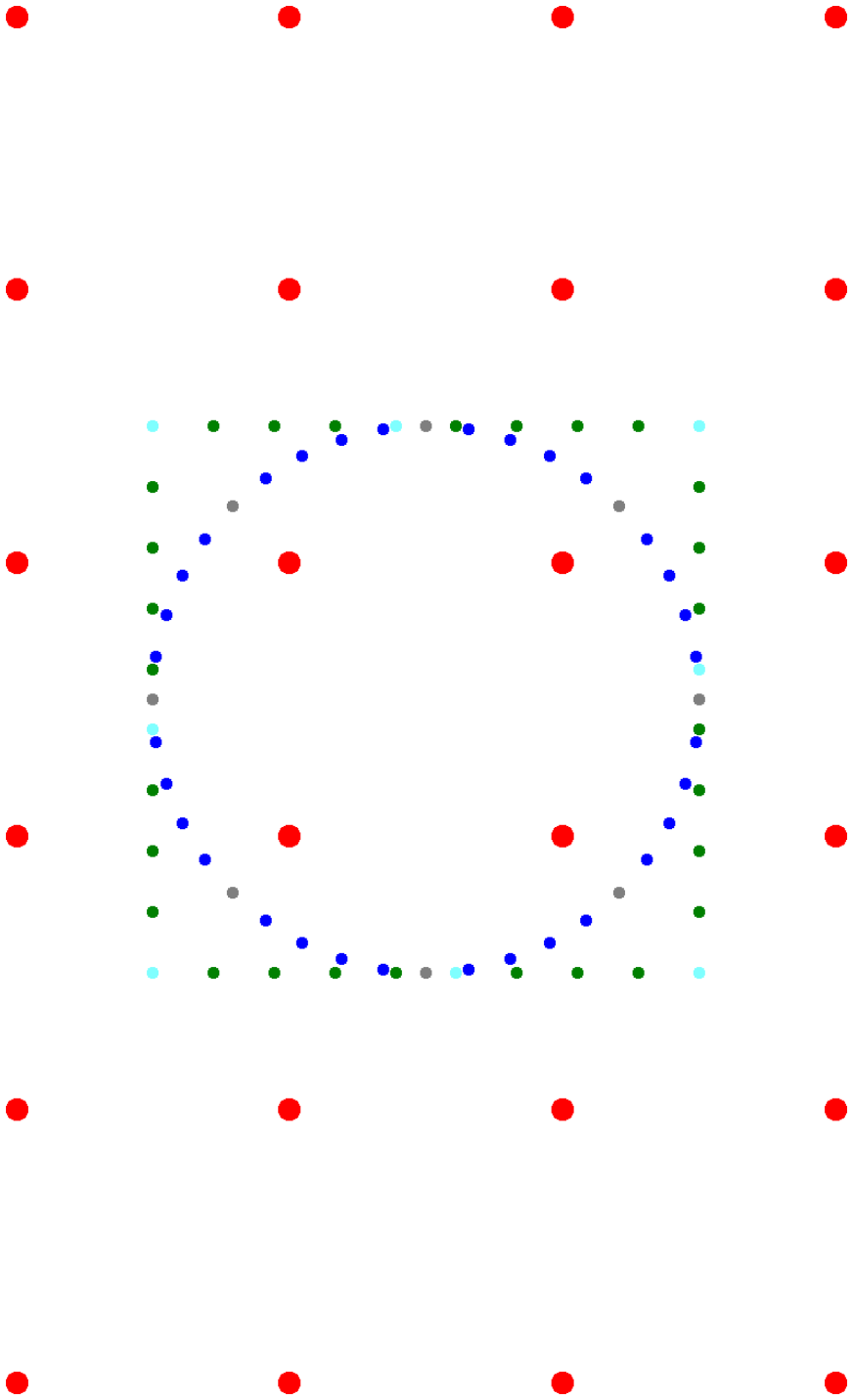


Figure 32. The rendered tactile points for the shape discrimination experiment. The red dots represent the tactors, the green dots are the rendered tactile points for the square stimulus, the blue dots are the rendered tactile points for the circle stimulus, the grey dots are the eight starting points for the circle stimulus, and the cyan dots are the eight starting points for the square stimulus.

4.2 Procedure

This section will outline the procedure each participant was taken through for the experiment. The setup of the experiment can be seen in Figure 33.

1) The participants were brought into the testing room.
2) The participants were presented with the consent form which they read over and signed.
3) Any questions the participants had were answered.
4) The participants were told that the experiment was composed of repeated trials of a tactile rendered stimulus of either a circle or a square and they would have to choose which one they believed they felt.
5) The participants then proceeded to put on the wearable tactile array.
6) The participants were then presented with a pair of headphones playing white noise and were told to adjust the volume to be more comfortable if necessary.
7) The Unity program was then started and the participants were told to click on the start experiment button when they were ready.
8) The participants were presented with the 160 stimuli discussed in the previous section.
9) Once the experiment was done the participants were presented with a short questionnaire to be filled out. (Appendix C)
10) The participants were thanked for their involvement and asked if they had any remaining questions.

Figure 33. Experiment setup.

5.0 Results & Discussion

This section will cover the results (Section 5.1) of the experiment described in Section 4 and discuss these results in detail (Section 5.2).

5.1 Results

A two-way repeated measures ANOVA was performed to determine three things; firstly, if the starting point of the rendered stimulus had an effect on the accuracy of detection, secondly, if the shape of stimulus had an effect on the accuracy of detection, and thirdly, if there is an interaction effect between the starting point and the shape. The ANOVA was performed using the Pingouin library for python (Vallat, 2018) for which you can see the output in Table 2. This analysis demonstrated that the starting point of the rendered shape had no effect on the accuracy of detection ($F = 0.32$, $p = 0.99$). It also demonstrated that the rendered shape had no effect on the accuracy of detection ($F = 0.18$, $p = 0.67$) and that there were no interaction effects between the starting point and the rendered shape on the accuracy of detection ($F = -0.01$, $p = 1.00$). The means of the detection accuracy for each starting point can be seen in Table 1. Since there were no differences in mean detection accuracy across starting points, the data for each starting point were merged for each shape.

Starting Point	Mean Detection Accuracy (%)
Circle 0	69.09
Circle 1	60.00
Circle 2	70.00
Circle 3	59.09
Circle 4	66.36
Circle 5	61.82
Circle 6	59.09
Circle 7	64.54
Square 0	63.64
Square 1	57.27
Square 2	65.54
Square 3	65.54
Square 4	66.36
Square 5	57.27
Square 6	62.73
Square 7	60.00

Table 1. The mean detection accuracy for each starting point of each shape.

Source	Sum of Squares	Degrees of Freedom	Mean Squares	F-Values	p-Values	Partial η^2 Effect Size
Start Point	0.263	15	0.018	0.318495	0.9929	0.032
Shape	0.010	1	0.010	0.181651	0.6706	0.001
Start Point * Shape	0.010	15	0.001	0.011959	1.0000	0.001
Residual	7.927	144	0.055			

Table 2. Two-way repeated measures ANOVA results.

In this experiment, the null hypothesis is that mean detection accuracy for each shape would not differ from chance which is 50%. This was tested using a one sample t-test using the SciPy library in Python (Virtanen et al., 2019) which demonstrated that the mean detection accuracy differed from 50% for both the circle stimulus (mean = 63.75%, $t = 2.6247$, $p < 0.05$) and the square stimulus (mean = 62.27%, $t = 2.2778$, $p < 0.05$) as well as the overall detection accuracy (mean = 63.01%, $t = 2.8054$, $p < 0.05$). The mean accuracy with 95% confidence intervals can be seen in Figure 34.

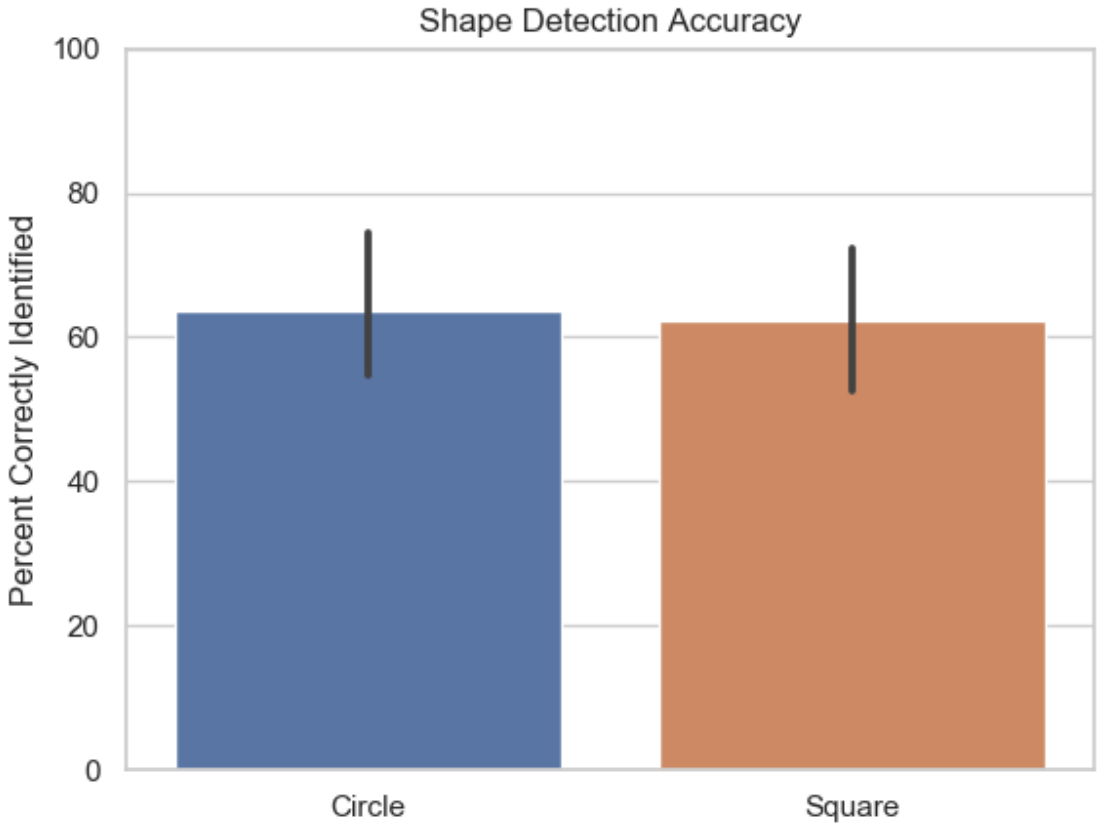


Figure 34. Shape detection accuracy.

Ten of the eleven participants rated the wearable tactile array as being comfortable with one rating it as being uncomfortable. For the question of "How well did you perceive the illusion?", a seven point likert scale was used with 1 being "Did not perceive the illusion", 4 being "Neutral", and 7 being "Perceived the illusion completely". The median score assigned by participants was 5, a histogram of scores can be seen in Figure 34.

A bivariate correlation between the likert scores and the shape detection accuracy scores was performed using Spearman's r, this was also performed using the SciPy library for python

(Virtanen et al., 2019). This resulted in non significant correlations for the circle stimuli ($r = 0.50$, $r^2 = 0.25$, $p = 0.12$), the square stimuli ($r = -0.34$, $r^2 = 0.12$, $p = 0.30$), and overall ($r = -0.07$, $r^2 = 0.005$, $p = 0.83$).

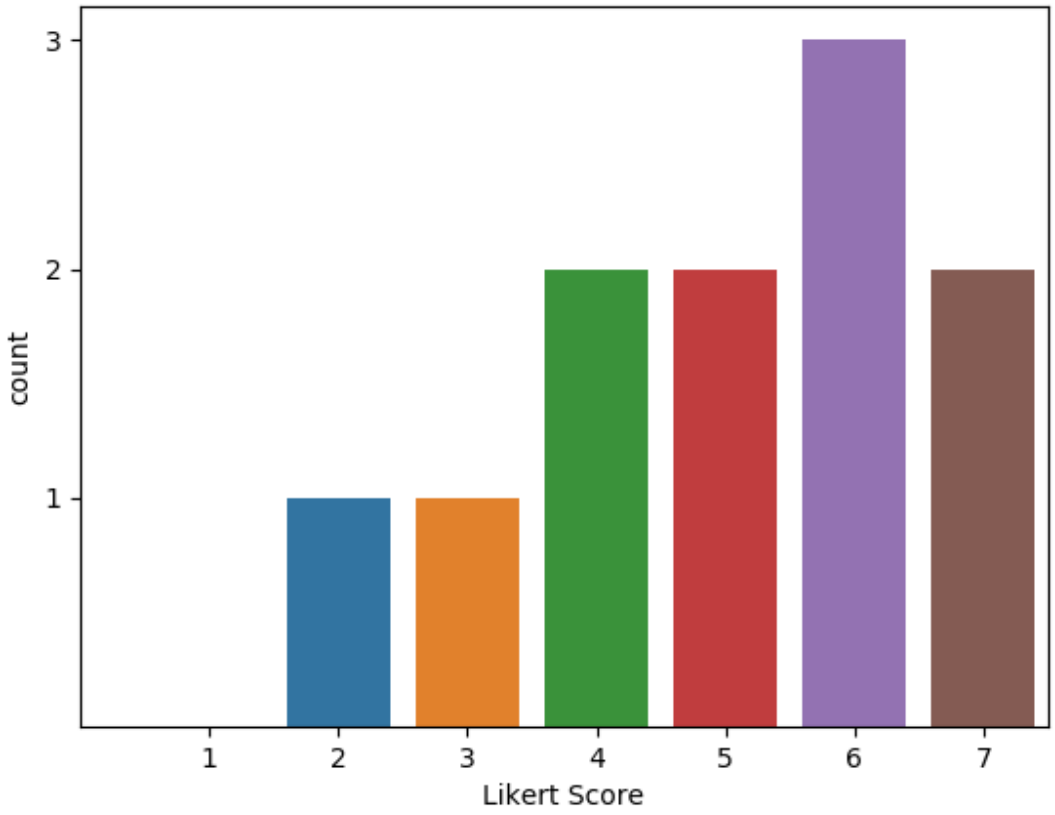


Figure 35. The distribution of likert scores for the question "How well did you perceive the illusion?"

5.2 Discussion

The starting point of the rendered shape had no effect on the accuracy of perception. This held true for both shapes and demonstrates that the location on the torso at which the shape starts being rendered has no effect on its perceptual clarity.

As described in the methodology, the two rendered shapes were crafted to be as similar as possible with the differentiating factor being the presence of a curve or corner which was rendered in a location that could only be done with the two dimensional funneling algorithm. Therefore, the mean detection accuracy for the rendered shapes were not expected to be high but were expected to differ from chance. This is the result that the experiment demonstrated with a mean detection accuracy of 63.75% for the circle stimulus, 62.27% for the square stimulus, and 63.01% overall which were all statistically significant findings. There was no significant difference of mean detection accuracy between the two stimuli which demonstrates that rounded and sharp stimuli are rendered as accurately using the two dimensional funneling algorithm.

Ten out of eleven participants found the wearable tactile array to be comfortable even though only one size was available which indicates that design requirement 2 (The wearable tactile array should feel safe and comfortable on the user) was met and that design requirement 4 (The wearable tactile array should fit various body sizes) was partially met.

As to how well the participants perceived the illusion, with an average likert score of 5 out of 7 with all participants above the 1 or “Did not perceive the illusion” option, this demonstrates that the participants perceived the illusion quite well. Although, it should be noted that the participants' subjective scores on how well they perceived the illusion did not correlate with their performance on the task.

6.0 Conclusion & Further Work

This thesis was undertaken with the goal of creating a wearable tactile array that was capable of exploiting the funneling illusion with a focus on a very usable design. A list of 20 requirements were developed based on previously established design guidelines and previous work done in haptic design. Only requirement 4, the wearable tactile array should fit various body sizes, was unable to be met.

The experiment that was performed demonstrated the feasibility of the novel two dimensional funneling algorithm. This algorithm is capable of rendering a dynamic tactile curved stimulus on the surface of the skin in between four equidistant tactors. This will allow for a richer tactile experience for users through an increase in the diversity of stimuli that is capable of being rendered on human skin.

The work done in this thesis can be applied in a variety of ways. The wearable tactile array allows for more immersive gaming experiences. For example, imagine you are playing a virtual reality sword fighting game; the wearable tactile array would allow the user to feel the slash crossing over their body in the same location on the torso as the in game avatar. Or, imagine a first person shooter where the exact location of the bullet hitting your avatar is felt leading to an increased awareness of the location of the enemy combatant. The rendering algorithm could be used in conjunction with a hand prosthesis device or robotic hand where the slippage of an object

out of the grasp of the device could be conveyed to the user to more quickly and naturally adjust the hand.

Future work that is possible is the improvement of the design to be able to fit various body sizes, especially women. Other work that can be done is to expand the SDK to include more tactile spatiotemporal illusions such as the cutaneous rabbit illusion, the tau effect, and the kappa effect. The motor PCBs could be further improved with the addition of H-bridges to allow for reverse braking which would allow for crisper stimuli.

References

Ackerley, R., Carlsson, I., Wester, H., Olausson, H., & Wasling, H.B. (2014). Touch perceptions across skin sites: differences between sensitivity, direction discrimination and pleasantness. *Frontiers in Behavioral Neuroscience, 8 (54)*, 1-10.

Alles, D. (1970). Information Transmission by Phantom Sensations. IEEE Transactions on Man Machine Systems, 11(1), 85-91.

Arendt-Nielsen, L., & Chen, A. (2003). Lasers and other thermal stimulators for activation of skin nociceptors in humans. *Neurophysiologie Clinique/Clinical Neurophysiology, 33(6)*, 259-268.

Ariza, O., Lubos, P., Steinicke, F., & Bruder, G. (2015). Ring-shaped haptic device with vibrotactile feedback patterns to support natural spatial interaction. *Proceedings of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments, ICAT - EGVE '15*, 175–181.

Ariza, O., Freiwald, J., Laage, N., Feist, M., Salloum, M., Bruder, G., & Steinicke, F. (2016). Inducing Body-Transfer Illusions in VR by Providing Brief Phases of Visual-Tactile Stimulation. *Proceedings of the 2016 Symposium on Spatial User Interaction*, 61-68.

Barghout, A., Cha, J., El Saddik, A., Kammerl, J. & Steinbach, E. (2009). Spatial Resolution of Vibrotactile Perception on the Human Forearm when exploiting Funneling Illusion. *2009 IEEE International Workshop on Haptic Audio Visual Environments and Games.*

Békésy, G.v. (1958). Funneling in the Nervous System and its Role in Loudness and Sensation Intensity on the Skin. The Journal of the Acoustical Society of America, 30 (5), 399 - 412.

Blankenburg, F., Ruff, C.C., Deichmann, R., Rees, G., & Driver, J. (2006). The Cutaneous Rabbit Illusion Affects Human Primary Sensory Cortex Somatotopically. PLoS Biology, 4(3), 459-466.

Cha, J., Eid, M., Rahal, L., & Saddik, A. E. (2008a). Hugme: An interpersonal haptic communication system. In Proc. of the Int. Conference on Haptic Audio visual Environments and Games, 99-102.

Cha, J., Rahal, L., & El Saddik, A. (2008b). A Pilot Study on Simulating Continuous Sensation with Two Vibrating Motors. *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 143-147.

Chen, L.M., Friedman, R.M., & Roe, A.W. (2003). Optical Imaging of a Tactile Illusion in Area 3b of the Primary Somatosensory Cortex. Science, 302 (5646), 881-885.

Cockburn, A., & Brewster, S. (2005). Multimodal feedback for the acquisition of small targets. Ergonomics, 48(9), 1129-1150.

Debus, T., Jang, T., Dupont, P., & Howe, R. (2002). Multi-channel vibrotactile display for teleoperated assembly. Proceedings 2002 IEEE International Conference on Robotics and Automation, 390-397.

Dinh, H.Q., Walker, N., Song, C., Kobayashi, A., & Hodges, L.F. (1999). Evaluating the Importance of Multi-sensory Input on Memory and the Sense of Presence in Virtual Environments. Proceedings of the IEEE virtual reality '99 conference, 222-228.

Fukuyama, K., Mizukami, Y., & Sawada, H. (2008). A Novel Micro-Vibrational Actuator and the Presentation of Tactile Sensations. 12th IMEKO TC1-TC7 Joint Symposium on Man, Science & Measurement, 141-146.

Gardner, E. P., & Spencer, W. A. (1972). Sensory funneling. I. Psychophysical observations of human subjects and responses of cutaneous mechanoreceptive afferents in the cat to patterned skin stimuli. Journal of Neurophysiology, 35(6), 925–953.

Gallace, A., Tan, H.Z., & Spence, C. (2007). The Body Surface as a Communications System: The State of the Art after 50 Years. *Presence: Teleoperators & Virtual Environments, 16 (6)*, 655-676.

Geldard, F.A. (1957). Adventures in tactile literacy. *American Psychologist, 12 (3)*, 115-124.

Geldard, F.A., & Sherrick, C.E. (1972). The Cutaneous "Rabbit": A Perceptual Illusion. *Science, 178 (4057)*, 178-179.

Gemperle, F., Kasabach, C., Stivoric, J., Bauer, M., & Martin, R. (1998). Design for Wearability. Proceedings of the ISWC 02, IEEE Computer Computer Society, 116-122.

Gemperle, F., Ota, N., & Siewiorek, D. (2001). Design of a wearable tactile display. Proceedings Fifth International Symposium on Wearable Computers, 5-12.

Giannopoulos, E., Pomes, A., & Slater, M. (2012). Touching the Void: Exploring Virtual Objects through a Vibrotactile Glove. The International Journal of Virtual Reality, 11(3), 19-24.

Gollner, U., Bieling, T., & Joost, G. (2012). Mobile Lorm Glove - Introducing a Communication Device for Deaf-Blind People. Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI 12, 127-130.

Gunther, E., & O'Modhrain, S. (2003). Cutaneous Grooves: Composing for the Sense of Touch. Journal of New Music Research, 32(4), 369-381.

Hawkes, G.R. (1960). Symposium on cutaneous sensitivity. *Medical Research Laboratories Report No. 424*. Fort Knox, USA.

Hein, A., & Brell, M. (2007). ConTACT - A Vibrotactile Display for Computer Aided Surgery. Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC07).

Ho, C., Tan, H. Z., & Spence, C. (2005). Using spatial vibrotactile cues to direct a driver's visual attention in driving scenes. Transportation Research Part F: Traffic Psychology and Behaviour, 8, 397-412.

Hsiao, S.S., Fitzgerald, P.J., Thakur, P.H., Denchev, P., & Yoshioka, T. (2009). Somatosensory Receptive Fields. *Encyclopedia of Neuroscience*, 111-119.

Hummel, J., Dodiya, J., Center, G. A., Eckardt, L., Wolff, R., Gerndt, A., & Kuhlen, T. W. (2016). A lightweight electrotactile feedback device for grasp improvement in immersive virtual environments. 2016 IEEE Virtual Reality (VR), 39-48.

Jay, C., Glencross, M., & Hubbold, R. (2007). Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. ACM Transactions on Computer-Human Interaction, 14(2):8.

Johansson, R.S. (1978). Tactile sensibility in the human hand: Receptive field characteristics of mechanoreceptive units in the glabrous skin area. The Journal of Physiology, 281(1), 101-125.

Johnson, K. (2001). The roles and functions of cutaneous mechanoreceptors. *Current Opinion in Neurobiology, 11(4)*, 455-461.

Jones, L.A., & Sarter, N.B. (2008). Tactile displays: guidance for their design and application. *Human Factors 50 (1)*, 90–111.

Kandel, E.R., Schwartz, J.H., Jessell, T.M., Siegelbaum, S.A., & Hudspeth, A.J. (2013). Principles of neural science. NY, NY: McGraw-Hill Medical.

Kato, H., Hashimoto, Y., & Kajimoto, H. (2010). Basic Properties of Phantom Sensation for Practical Haptic Applications. Haptics: Generating and Perceiving Tangible Sensations Lecture Notes in Computer Science, 271-278.

Kennett, S., Spence, C., & Driver, J. (2002). Visuo-tactile links in covert exogenous spatial attention remap across changes in unseen hand posture. Perception & Psychophysics, 64 (7), 1083-1094.

Kerdegari, H., Kim, Y., Stafford, T., & Prescott, T. J. (2014). Centralizing Bias and the Vibrotactile Funneling Illusion on the Forehead. Haptics: Neuroscience, Devices, Modeling, and Applications Lecture Notes in Computer Science, 8619, 55-62.

Kilgard, M.P. & Merzenich, M.M. (1995). Anticipated stimuli across skin. Nature, 373 (6515), 663.

Kim, D., Lu, N., Ma, R., Kim, Y., Kim, R., Wang, S., Wu, J., Won, S.M., Tao, H., Islam, A., Yu, K.J., Kim, T., Chowdhury, R., Ying, M., Xu, L., Li, M., Chung, H., Keum, H., McCormick, M., Liu, P., Zhang, Y., Omenetto, F.G., Huang, Y., Coleman, T., & Rogers, J. A. (2011). Epidermal Electronics. Science, 333(6044), 838-843.

Kim, K. & Colgate, J.E. (2012). Haptic Feedback Enhances Grip Force Control of sEMG-Controlled Prosthetic Hands in Targeted Reinnervation Amputees. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 20 (6), 798-805.

Kuroki, S., Kajimoto, H., Nii, H., Kawakami, N., & Tachi, S. (2007). Proposal for tactile sense presentation that combines electrical and mechanical stimulus. Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC07).

Lee, J., Kim, Y., & Kim, G.J. (2012). Funneling and Saltation Effects for Tactile Interaction with Virtual Objects. *CHI'12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3141 - 3148.

Lindeman, R.W., & Cutler J.R. (2003). Controller Design for a Wearable, Near-Field Haptic Display. *Proc. of the 11th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 397-403.

Lindeman, R.W., Page, R., Yanagida, Y., & Sibert, J.L. (2004). Towards Full-Body Haptic Feedback: The Design and Deployment of a Spatialized Vibrotactile Feedback System. Proceedings of the ACM symposium on Virtual reality software and technology, 146-149.

Michelson, J.D., & Hutchins, C. (1995). Mechanoreceptors in Human Ankle Ligaments. *The Journal of Bone & Joint Surgery, British Volume, 77-B(2)*, 219-224.

Miyazaki, M., Hirashima, M., & Nozaki, D. (2010). The "Cutaneous Rabbit" hopping out of the body. The Journal of Neuroscience, 30 (5), 1856-1860.

Mizukami, Y. & Sawada, H. (2006). Tactile information transmission by apparent movement phenomenon using shape-memory alloy device. Proc. 6th Intl Conf. Disability, Virtual Reality & Assoc. Tech., 133-140.

Moriyama, T.K., Nishi, A., Sakuragi, R., Nakamura, T., & Kajimoto, H. (2018). Development of a wearable haptic device that presents haptics sensation of the finger pad to the forearm. 2018 IEEE Haptics Symposium (HAPTICS), 180-185.

Munger, B.L., & Ide, C. (1988). The Structure and Function of Cutaneous Sensory Receptors. *Arch. Hist. Cytol., 51 (1)*, 1-34.

Muramatsu, Y., Niitsuma, M., & Thomessen, T. (2012). Perception of tactile sensation using vibrotactile glove interface. 2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), 621-626.

Oohara, J., Kato, H., Hashimoto, Y., & Kajimoto, H. (2010). Presentation of Positional Information by Heat Phantom Sensation. *Haptics: Generating and Perceiving Tangible Sensations Lecture Notes in Computer Science*, 445-450.

Pawling, R., Cannon, P.R., McGlone, F.P., & Walker, S.C. (2017). C-tactile afferent stimulating touch carries a positive affective value. PLoS ONE, 12(3): e0173457.

Piateski, E., & Jones, L.A. (2005). Vibrotactile Pattern Recognition on the Arm and Torso, Proceedings of Worldhaptics.

Rahal, L., Cha, J., Saddik, A.E., Kammerl, J., & Steinbach, E. (2009a). Investigating the Influence of Temporal Intensity Changes on Apparent Movement Phenomenon. IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems, 310-313.

Rahal, L., Cha, J., & Saddik, A.E. (2009b). Continuous tactile perception for vibrotactile displays. 2009 IEEE International Workshop on Robotic and Sensors Environments, 86-91.

Relaxing White Noise (2015, April 28). CELESTIAL WHITE NOISE | Sleep Better, Reduce Stress, Calm Your Mind, Improve Focus | 10 Hour Ambient [video file]. Retrieved from https://www.youtube.com/watch?v=wzjWIxXBs_s

Richter, Hendrik & Blaha, Benedikt & Wiethoff, Alexander & Baur, Dominikus & Butz, Andreas. (2011). Tactile Feedback without a Big Fuss: Simple Actuators for High-Resolution Phantom Sensations. UbiComp'11 - Proceedings of the 2011 ACM Conference on Ubiquitous Computing. 85-88.

Rupert, A.H., Lawson, B.D., & Basso, J.E. (2016). Tactile Situation Awareness System: Recent Developments for Aviation. Proceedings of the Human Factors and Ergonomics Society 2016 Annual Meeting, 722-726.

Schepers, R.J., & Ringkamp, M. (2009). Thermoreceptors and thermosensitive afferents. *Neuroscience & Behavioural Reviews, 33 (3)*, 205-212.

Sherrick, C.E., Cholewiak, R.W., & Collins, A.A. (1990). The localization of low-and high-frequency vibrotactile stimuli. J. Acoust. Soc. Am., 88 (1), 169-179.

Sofia, K. O., & Jones, L. (2013). Mechanical and Psychophysical Studies of Surface Wave Propagation during Vibrotactile Stimulation. IEEE Transactions on Haptics, 6(3), 320-329.

Spagnoletti, G., Tommaso, L.M., Baldi, L., Gioioso, G., Pacchierotti, C., & Prattichizzo, D. (2018). Rendering of pressure and textures using wearable haptics in immersive VR environments. *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*.

Tachi, S., Tanie, K., Komoriya, K., & Abe, M. (1985). Electrocutaneous Communication in a Guide Dog Robot (MELDOG). IEEE Transactions on Biomedical Engineering, BME-32(7), 461-469.

Tang, H., & Beebe, D. (2006). An Oral Tactile Interface for Blind Navigation. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 14(1), 116-123.

Vallat, R. (2018). Pingouin: statistics in Python. *Journal of Open Source Software, 3(31)*, 1026.

Van Erp, J.B.F., Veltman, J.A., Van Veen, H. A. H. C., & Oving, A. B. (2002). Tactile Torso Display as Countermeasure to Reduce Night Vision Goggles Induced Drift, NATO RTO Conference on Spatial Disorientation, 15-17.

Verrillo, R.T. (1966). Vibrotactile Thresholds for Hairy Skin. Journal of Experimental Psychology, 72 (1), 47-50.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., & SciPy 1.0 Contributors. (2019). SciPy 1.0-Fundamental Algorithms for Scientific Computing in Python. preprint arXiv:1907.10121

Yanagida, Y., Kakita, M., Lindeman, R.W., Kume, Y., & Tetsutani, N. (2004). Vibrotactile Letter Reading Using a Low-Resolution Tactor Array. Proc. of the 12th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 400-406.

Yang, U., Jang, Y., & Kim, G.J. (2002). Designing a Vibro-Tactile Wear for “Close Range” Interaction for VR-based Motion Training. Proceedings of the International Conference on Artificial Reality and Telexistence, 4-9.

Ying, M., Bonifas, A. P., Lu, N., Su, Y., Li, R., Cheng, H., Ameen, A., Huang, Y., & Rogers, J. A. (2012). Silicon nanomembranes for fingertip electronics. *Nanotechnology, 23(34)*, 344004.

Appendix A

Motor	Current Draw (mA)
1	87.9
2	94.8
3	93.5
4	100
5	88.7
6	93
7	90.6
8	94.3
9	92.9
10	91.9
11	91.8
12	93.7
13	90.6
14	89.6
15	93.8
16	90.2
17	103.1
18	90.8
19	95.3

20	95.2
21	91.9
22	93
23	89.2
24	91.5
25	91.9
26	92.8
27	90.5
28	92.9
29	90.1
30	90.3
31	96
32	89
33	92.9
34	95
35	88
36	94.5
37	92.9
38	88
39	93.4
40	84.3
41	83.3
42	85
43	88.9
44	97.2
45	100.4
46	95.6
47	91.9
48	92.9
49	92.2
50	89.2
51	91.5
52	90.5
53	93.2

54	92.4
55	93.2
56	84.3
57	91.7
58	94.7
59	90
60	87.7
61	93.6
62	91.9
63	89.8
64	90.3
65	93.2
66	90.3
67	89.8
68	86.2
69	94
70	90.9
71	94.5
72	92
73	93.8
74	89.9
Mean	91.84
Std. Dev.	3.41

		1	2	3	4	5	6	Mean	Std. Dev.
Duty Cycle	**0%**	7.2	7.2	7.3	7.3	7.4	7.3	7.3	0.075
	24.60%	103.7	99.3	103.6	101.1	103	103.8	102.4	1.830
	50%	204.3	200.3	201.6	199.1	198.3	189.6	198.9	5.006
	75.40%	288.5	289.4	296.3	294.4	286.5	285.8	290.2	4.276
	100%	376	382.3	386.2	382.5	377	375.2	379.9	4.425

Appendix B

SDK Documentation

TactileObject

TactileObject

public TactileObject(string objectName)

public TactileObject(int[] coordinateArray, string objectName, double speed, double distance, bool front = true)

Parameters

objectName	The name of the TactileObject.
coordinateArray	An int array stucture as [x,y,x,y,x,y,....,x,y]. The value of x must be between 0 and 299 and the value of y must be between 0 and 499.
speed	The speed at which the funneling illusion will propagate in mm/s.
distance	The intermotor distance in the array being used.
front	A boolean which states if this TactilePoint is to be displayed at the front of the HapSys vest. Default is true.

Description

Constructor which creates an empty TactileObject of the given name or a constructor which creates a TactileObject composed of TactilePoints whose coordinates are given in coordinateArray.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;
        private TactileObject tactObj;
```

```
	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

addTactilePoint

public void addTactilePoint(TactilePoint tp)

Parameters

tp The TactilePoint to be added to the TactileObject.

Description

Function which adds the given TactilePoint to the TactileObject.

Usage

```
using UnityEngine;
using HapSysSDK;
```

```
public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObject tactObj;

	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

changeObjectName

public void changeObjectName(string newObjectName)

Parameters

newObjectName	The new name of the TactileObject.

Description

Function which renames the TactileObject.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObject tactObj;

	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

getObjectName

```
public string getObjectName()
```

Description

Function which returns the name of the TactileObject.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObject tactObj;

	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

getTactilePoint

```
public TactilePoint getTactilePoint(int index)
```

Parameters

index	The position of the TactilePoint within the TactileObject. The first TactilePoint is at index 0.

Description

Function which returns the TactilePoint found at the given index of the TactileObject.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObject tactObj;

	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

getSize

```
public int getSize()
```

Description

Function which returns the number of TactilePoints which compose the TactileObject.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObject tactObj;

	void Start() {
		vest = new Vest (“com8”);
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		tactObj.changeObjectName("Corners");
		Debug.Log("Tactile Object name is " + tactObj.getObjectName());
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactilePoint tp = tactObj.getTactilePoint(Random.value *
(tactObj.getSize() - 1));
			vest.sendTactilePoint(tp);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

TactileObjectDictionary

TactileObjectDictionary

public TactileObjectDictionary()

Description

Constructor which creates a dictionary which is capable of storing TactileObjects and retrieving them by name.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;
	private TactileObjectDictionary dict;

	void Start() {
		vest = new Vest (“com8”);
		dict = new TactileObjectDictionary ();
		tactObj = new TactileObject ("tactObj");
		tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
		tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
		tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
		dict.addTactileObject(tactObj)
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			TactileObject tactObj = dict.getTactileObject("tactObj");
			vest.sendTactileObject(tactObj);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

addTactileObject

public void addTactileObject(TactileObject tactObj)

Parameters

tactObj	The TactileObject to be added to the TactileObjectDictionary.

Description

Function which adds the given TactileObject to the TactileObjectDictionary.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;
        private TactileObjectDictionary dict;

        void Start() {
                vest = new Vest (“com8”);
                dict = new TactileObjectDictionary ();
                tactObj = new TactileObject ("tactObj");
                tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
                tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
                tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
                tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
                dict.addTactileObject(tactObj)
        }

        void Update() {
                if (Input.getMouseButtonDown(0)) {
                        TactileObject tactObj = dict.getTactileObject("tactObj");
                        vest.sendTactileObject(tactObj);
                }
        }

        void OnApplicationQuit() {
                vest.closePort ();
        }
```

}

getTactileObject

public TactileObject getTactileObject(string tactObjName)

Parameters

tactObjName	The name of the TactileObject to be retrieved from the TactileObjectDictionary.

Description

Function which returns the TactileObject of the given name from the TactileObjectDictionary.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;
        private TactileObjectDictionary dict;

        void Start() {
                vest = new Vest (“com8”);
                dict = new TactileObjectDictionary ();
                tactObj = new TactileObject ("tactObj");
                tactObj.addTactilePoint(new TactilePoint (0, 0, 1000));
                tactObj.addTactilePoint(new TactilePoint (299, 0, 1000));
                tactObj.addTactilePoint(new TactilePoint (0, 499, 1000));
                tactObj.addTactilePoint(new TactilePoint (299, 499, 1000));
                dict.addTactileObject(tactObj)
        }

        void Update() {
                if (Input.getMouseButtonDown(0)) {
                        TactileObject tactObj = dict.getTactileObject("tactObj");
                        vest.sendTactileObject(tactObj);
                }
        }

        void OnApplicationQuit() {
```

```
            vest.closePort ();
        }
}
```

TactilePoint

TactilePoint

public TactilePoint(int x, int y, float duration, bool front = true)

Parameters

x	The x coordinate of the TactilePoint. Value must be between 0 and 299.
y	The y coordinate of the TactilePoint. Value must be between 0 and 499.
duration	The duration for which the TactilePoint will be displayed in seconds.
front	A boolean which states if this TactilePoint is to be displayed at the front of the HapSys vest. Default is true.

Description

Constructor which creates the TactilePoint at the given x and y coordinates which last for the given duration.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;

        void Start() {
                vest = new Vest ("com8");
                TactilePoint tp = new TactilePoint (150, 250, 1.0F);
        }

        void OnApplicationQuit() {
                vest.closePort ();
        }
```

}

changeDuration

public void changeDuration(float newDuration)

Parameters

newDuration	The duration for which the TactilePoint will be displayed in seconds.

Description

Function which updates the duration for which the TactilePoint will be displayed.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;

        void Start() {
                vest = new Vest (“com8”);
                TactilePoint tp = new TactilePoint (150, 250, 1.0F);
                tp.changeDuration(2.0F);
        }

        void OnApplicationQuit() {
                vest.closePort ();
        }
}
```

sendPoint

public void sendPoint(SerialPort sp)

Description

Function used by the Vest class to facilitate sending the TactilePoint to the HapSys Vest.

Vest

Vest

public Vest(string com)

Parameters

com	The com port which is being used by the HapSys tactile vest

Description

Constructor which creates the vest object that contains all the methods needed to interface with the HapSys tactile vest and opens the COM Port.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;

        void Start() {
                vest = new Vest (“com8”);
        }

        void OnApplicationQuit() {
                vest.closePort ();
        }
}
```

activateMotors

public void activateMotors(int[] motorList, float duration)

Parameters

motorList	Array of motors to be activated. The motors on the front of the vest are listed 0 - 23 and the motors on the back of the vest are listed 24 - 47 as

seen on the diagram below.

duration	The duration the motors are to be activated in seconds.

Description

This function activates the motors listed in motorList for duration seconds.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			int[] randomMotors = new int[(int) (Random.value * 47)];
			for(int i = 0; i < randomMotors.Length; i++) {
				randomMotors[i] = i;
			}
			vest.activateMotors(randomMotors, 1.0F);
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

Diagram

addTactileObjectToDict

public void addTactileObjectToDict(TactileObject tactObj)

Parameters

tactObj	The TactileObject to be added to the dictionary.

Description

This function adds the TactileObject to the dictionary so it can be easily accessed via the getTactileObject function.

Usage

```
using UnityEngine;
using HapSysSDK;
```

```
public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
		int[] array = new int[20];
		for(int i = 0; i < array.Length; i++) {
			array[i] = i * 10;
		}
		TactileObject to = new TactileObject (array, "Line");
		vest.addTactileObjectToDictionary (to);
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

closePort

public void closePort()

Description

Function which closes the COM Port connecting the HapSys vest.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

getTactileObject

public TactileObject getTactileObject(string name)

Parameters

name	The name of the TactileObject.

Description

This function retrieves the TactileObject of the given name from the TactileObjectDictionary.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
		int[] array = new int[20];
		for(int i = 0; i < array.Length; i++) {
			array[i] = i * 10;
		}
		TactileObject to = new TactileObject (array, "Line");
		vest.addTactileObjectToDictionary (to);
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			sendTactileObject(vest.getTactileObject("Line"));
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

sendLine

public void sendLine(int startX, int startY, int endX, int endY, double speed = 60.0, bool front = true)

Parameters

startX	The x coordinate of the starting point. Value must be between 0 and 299.
startY	The y coordinate of the starting point. Value must be between 0 and 499.
endX	The x coordinate of the ending point. Value must be between 0 and 299.
endY	The y coordinate of the ending point. Value must be between 0 and 499.
speed	The speed at which the line will propagate.
front	A boolean which states if this TactileObject is to be displayed at the front of the HapSys vest. Default is true.

Description

This function creates and sends a line of pressure to the vest which starts at point (startX, startY) and ends at point (endX, endY).

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
		vest.sendLine (0, 0, 299, 499);
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

sendTactileObject

public void sendTactileObject(TactileObject tactObj)

Parameters

tactObj	The TactileObject to be displayed on the HapSys Vest.

Description

This function sends the given TactileObject to the HapSys Vest to be displayed.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
	private Vest vest;

	void Start() {
		vest = new Vest (“com8”);
		int[] array = new int[20];
		for(int i = 0; i < array.Length; i++) {
			array[i] = i * 10;
		}
		TactileObject to = new TactileObject (array, "Line");
		vest.addTactileObjectToDictionary (to);
	}

	void Update() {
		if (Input.getMouseButtonDown(0)) {
			sendTactileObject(vest.getTactileObject("Line"));
		}
	}

	void OnApplicationQuit() {
		vest.closePort ();
	}
}
```

sendTactilePoint

public void sendTactilePoint(TactilePoint tp)

Parameters

tp	The TactilePoint to be displayed on the HapSys Vest.

Description

This function sends the given TactilePoint to the HapSys Vest to be displayed.

Usage

```
using UnityEngine;
using HapSysSDK;

public class gameVest : MonoBehaviour {
        private Vest vest;

        void Start() {
                vest = new Vest (“com8”);
                TactilePoint tp = new TactilePoint (150, 250, 1000);
                vest.sendTactilePoint(tp);
        }

        void OnApplicationQuit() {
                vest.closePort ();
        }
}
```

Appendix C

Subject ID: _____

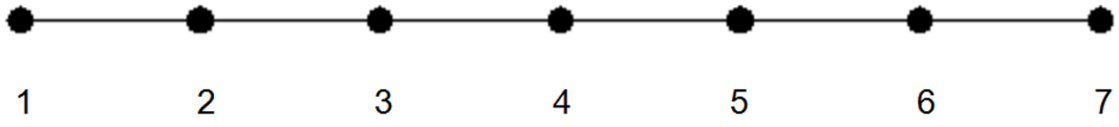
Age

- 18 - 24
- 25 - 29
- 30 - 39
- 40 - 49
- 50 - 59

Questionnaire

How well did you perceive the illusion:

Did not perceive the illusion — Neutral — Perceived the illusion completely



How did you find the vest?

- Comfortable
- Uncomfortable
- Don't Know