Haptic Navigation

Developing Eyes-Free Pedestrian Navigation on Apple Watch

Author: Gabriel Frederik Brodersen (gbro@itu.dk)
Education: MSc in Software Design
Project Code: KISPECI1SE
Date: 1st of September 2021

IT UNIVERSITY OF COPENHAGEN

Title: Haptic Navigation: Developing Eyes-Free Pedestrian Navigation on Apple Watch

Author: Gabriel Frederik Brodersen (gbro@itu.dk)

Project: Master Thesis

Project Code: KISPECI1SE

Extent: 30 ECTS

Date: 1st of September 2021

Programme: Software Design

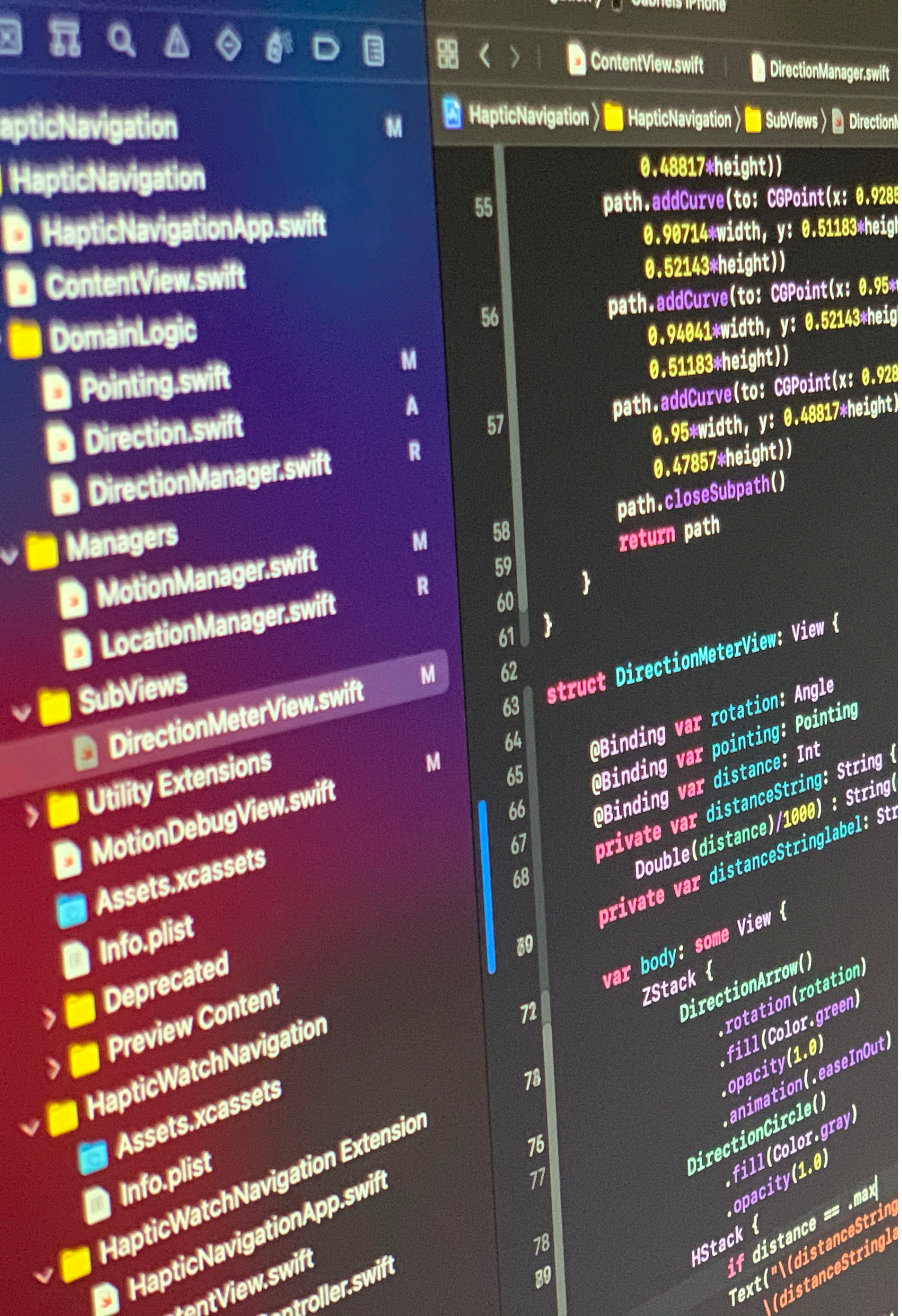
University: IT University of Copenhagen, Denmark

Supervisor: Gopinaath Kannabiran (goka@itu.dk)

Abstract: Typical pedestrian navigation systems available on modern smartphones today still use the same map-based navigation paradigm adopted from car navigation. This is although traveling on foot and in traffic, presents considerably different challenges in terms of safety and efficiency. Walking in traffic while looking at a screen to read a map is a potential safety risk. Turn-by-turn navigation doesn't represent the diversity and flexibility of options available to pedestrians. Modern smartwatches usually copy this approach of screen-based navigation, although displayed on a smaller screen size. This project presents the design and development of a novel multimodal interface for pedestrian navigation that facilitates visual and eye-free feedback using haptic feedback on the wrist. The prototype uses location and motion tracking to track the user's location and movement. This report describes the design and development considerations made during the project. Participants were used as testers and observed using the prototype in a realistic context to evaluate the final design. The final prototype consists of an Apple Watch and companion iPhone app that the participants could install on their own devices during in-person testing and to provide ongoing feedback during the span of the project.

Keywords: Smartwatch, Pedestrian Navigation, Way-finding, Haptic Interfaces, Bearing-based Navigation

Prototype Download Link
https://testflight.apple.com/join/45iwXw1N
Requires TestFlight and a paired Apple Watch series 5 or newer



Prototype Source Code

https://github.com/Brodersen/HapticNavigation

GitHub Repository

Table of Contents

Introduction **6**
Related Work **7**
Challenges with Pedestrian Navigation 7
An Opportunity for Haptics 9
An Opportunity for Smartwatches 10
Technical Challenges of Today's Navigation Technology 11
Related Projects 11
Methodology **13**
Testing 14
Design **16**
Using a Digital Compass for Directional Guidance 16
Determining the Direction of the User 17
Using Haptics for Directional Guidance 19
Using Haptics for Distance Sensing 22
Creating Eyes-Free Navigation 22
The Final Prototype: Haptic Navigator **24**
A Tool for Wayfinding 24
Orientation 25
Route Selection 27
Route Control 28
Recognition of Destination 29
Sensing Direction as a Reaction to Interaction 30
Taking Advantage of the Smartphone 31
Development **33**
Hardware Considerations 33
Software Development 33
Software Architecture 37
Platform Limitations 39
Haptic API Limitations 39
Findings **42**
User Feedback and Observations 42
Learning Curve 42
Eyes-Free Navigation 43
Social Acceptance 44
Personal Preferences 45
Technical Observations 46
Conclusion **47**
Bibliography **48**

Introduction

Ever since the invention of Global Positioning Systems (GPS) made its way from car navigation to personal navigation services and apps on smartphones, the idea of visually reading a map on screen and listening to spoken turn-by-turn directions has been the predominant approach to guiding the user to their chosen destination. Although turn-by-turn navigation is suitable for vehicle navigation driving on clearly defined roads, this same navigation paradigm presents interesting challenges when used in a pedestrian navigation context (Gartner et al., 2011; Pielot et al., 2010a; Renaudin et al., 2017). Several attempts have been made to develop alternative solutions to pedestrian navigation that are both more convenient, practical, and safer to use in traffic than classic turn-by-turn navigation. A considerable subset of these solutions utilize haptic technology in wearables as an attempt to provide eyes-free navigation that allow the user to keep their attention on the road and avoid distractions from screen-based navigation.

The hardware used in many of these projects, such as GPS, motion-tracking sensors, and haptic technology, are now common additions to commercially available smartwatches. Some of these smartwatches are open for software developers to develop creative solutions that utilize these hardware capabilities, combined with the wearability and feasibility offered by smartwatches. The use of modern smartwatches presents a suitable candidate to explore potential solutions to current challenges with screen-based pedestrian navigation systems. The presented project is exploratory, presenting the design, development, and testing of a wrist-worn prototype utilizing haptic feedback to improve the human-computer interaction (HCI) involved in pedestrian navigation. The project examines the challenges of pedestrian navigation in relation to car navigation since the dominant solutions still used today are derived from the same map-based navigation paradigm.

The use of a modern smartwatch is considered for its benefits as a conduit for haptic feedback being in direct contact with the user's skin. The smartwatch also presents an interesting opportunity as a motion tracking device, for its placement near the hand that is at the same time easily accessible and appropriate as a wearable. The project consists mainly of the design and development of a prototype aimed to answer the following research question:

"How can haptic technology be utilized in a smartwatch for eyes-free pedestrian navigation?"

The final prototype delivered consists of an Apple Watch and iPhone app bundle, chosen as the platform of choice.

Findings from tests conducted with participants are described, presenting relevant observations, proof of concept tests, and key insights gathered from observing participants and potential end-users using the app on their own devices and for navigating in realistic and public urban and suburban settings.

Related Work

The following section describes problems with current navigation systems for pedestrians that are not solved by adopting car navigation. A case is made for exploring haptics as an opportunity to address the challenges of pedestrian navigation and how the popularity of modern smartwatches presents a timely platform for design exploration. Finally, related projects are explored within the area of pedestrian navigation, and the design of haptic interfaces to solve navigation challenges are explored, providing the foundation for this project.

This project is inspired by previous work done by the author, exploring the use of vibrotactile feedback on the wrist as an interface to communicate directional information (Brodersen, 2019).

Challenges with Pedestrian Navigation

Navigating on foot typically requires the user to hold a smartphone in hand in order to read the on-screen map from a navigation app, thereby taking their attention away from the road and traffic. To avoid walking into something while looking at the screen, a person might stop walking while looking and only start walking again when not looking at the screen. To interpret the information presented on a screen, such as a user's location and the direction of the roads and suggested route, requires that the user mentally orientate the map's frame of reference to their own frame of reference in the real world. This transformation of the frame of reference can cause a noticeable cognitive load depending on how much rotation is needed to relate the map with the real world (O'Modhrain, 2004). Depending on ones ability to interpret a map, the rotation problem can be remedied by the user rotating the map or turning around themselves in order to line up the direction of the map with their own direction in the world, reducing the cognitive load of interpreting their presence and orientation on the digital map (Laurier & Brown, 2008). With car navigation, this map is typically automatically rotated to fit the car's driving direction, eliminating the need to perform any mental rotation.

Navigating using a hand-held map also means that the user must occasionally take their attention away from their surroundings in order to interpret the map and the progress of their navigation, posing a potential safety risk (Hyman Jr et al., 2010; Pielot et al., 2011a). This can be more challenging if the screen is hard to read, such as in direct sunlight or in bad weather. Hand-held navigation also requires that the user have one or both hands free to hold and interact with the smartphone, making the interaction unsuitable while carrying objects or wearing gloves. The user can temporarily put away the smartphone to avoid holding it in hand or exposing it to environmental conditions, but this also makes the state of navigation temporarily unavailable.

Another option is to use audio cues, such as sound alerts and spoken navigation instructions, as common with car navigation, to assist the map navigation and allow the driver to keep their eyes on the road. If sound navigation should be used in a walking context, headphones are usually preferred to keep the information private and to avoid disturbing others nearby. However, wearing headphones and using voice navigation while walking in traffic poses the safety risk of not paying attention to one's surroundings (Lichenstein et al., 2012). Headphones can also be inconvenient to use while navigating with others. These problems of privacy and paying attention to traffic is less prevalent while driving, where navigation can only be heard by the people inside the car, and where shielding out ambient noise can be a preferred feature of the car.

In terms of navigation accuracy, car navigation also has considerable advantages to pedestrian navigation. The geolocation and direction of the car can be estimated by associating the location and direction with data about nearby roads, as cars will likely be on a mapped road. Since the location and direction of cars can be tied to the map data of the road on which they are driving, and since speed and acceleration is often relatively predictable on roads, the system can use this information to estimate the state of the navigation to closely match the reality under typical driving conditions.

To change course while driving is also typically limited to the next upcoming intersection or junction. Although the same predictions can be applied to pedestrian navigation when walking on roads, there are some noteworthy differences. First, traveling on foot means that the user is often exposed to more options for changing their routes. Their options are typically the same as cars, but with the added freedom of walking off the road and taking paths that are inaccessible to cars. These include taking shortcuts and cutting corners, walking on unmapped roads, hiking paths, walking over open terrain such though parks, playgrounds, plazas, vacant lots, fields, town squares, forests, etc. Entering and passing through public buildings like a train station or mall with multiple exits is also an option and can compromise the quality of the GPS signal. Second, the direction of walking can be less predictable since a person is free to make spontaneous changes in their walking direction and can start or stop at any point — something that is not always possible (safe or legal) to do in a car mid-driving on roads.

Measuring the direction of a person is also a challenge compared to the direction of a car. A car can have hardware censored fixed in place in the car’s frame, and the direction is often one way or the other on the road on which the car is driving on. In contrast, a person might have their smartphone stored in a pocket, a bag, or held in their hands, which means that rotation and direction of the smartphone relative to the direction the person is facing cannot be guaranteed. These challenges also mean that navigation instructions using voice commands suitable for car navigation are not always compatible for use in pedestrian navigation. For example, "Turn Right" can mean many things when walking through an open park or town square. "Take the next left turn" can be ambiguous or confusing when the pedestrian is exposed to multiple upcoming paths that might not all be accounted for on a map or can mean different angles of directions than just 'left'.

The current challenges with pedestrian navigation on everyday mobile devices present an interesting design space for exploring alternative futures of how these devices can be used and improved to make the navigation experience more convenient, accessible, and safe.

An Opportunity for Haptics

The previous section examined how the visual and auditory modalities are used in pedestrian navigation using a map-based approach. A third modality often underappreciated in the design of digital artifacts is the sense of touch, which can be distinguished as three types of sensory experiences: Cutaneous, kinesthetics, and haptic (Biswas & Visell, 2021). The haptic perception in particular — which can be stimulated using haptic technology by applying forces, vibrations, or motions to the user — is the most widely used in digital devices today. This is likely due to the ease of creating vibrotactile stimulation using small vibrating motors that can fit into many types of mobile electronic consumer products. In smartphones, the use of haptic is often used to provide silent feedback about messages or incoming phone calls by vibrating the phone.

Depending on the capability of the technology used, haptic feedback can be expressed by various patterns with different sharpness, intensity, and rhythm, to express different types of information. These patterns can include anything from transient experiences such as brief clicks, taps or impulses, to continuous events like sustained vibrations and buzzing. By varying sharpness and intensity, haptics can feel soft and rounded, sharp and mechanical, or anything in-between. Haptic Icons can be created by associating meaning to different haptic patterns and using them consistently throughout the interface in which they are used (Maclean & Enriquez, 2003). For example, by associating visual information — such as clicking a failure or success button in the user interface of an app, with corresponding haptic icons like a buzzing pattern (for failure) and quick taps (for success) — the user can create associations between the haptic information and the visual information presented, connecting the two sensory experiences.

Haptic feedback is interesting because it provides unique advantages and disadvantages compared to other interactive modalities (Sreelakshmi & Subash, 2017). For one, haptic interfaces can be used to provide eyes-free information to the user, as long as the emitted feedback can be felt by the user’s skin. This type of feedback can also be silent enough to provide a private information delivery that other people cannot see or hear while also being socially convenient by avoiding disturbing others. Another benefit comes from situations where the user’s visual or auditory attention is already occupied. In these situations, interfaces that require attention using the same sensory channels can lead to attentive conflicts, fragmented attention, and decreased performance (Barnard et al., 2006; Oulasvirta et al., 2005). Instead of providing multiple sources of feedback on the same sensory channels already occupied, an alternative approach is to explore the use of other sensory channels that are less stimulated. Spreading out information feedback to the other human senses using multimodal interfaces to appeal to senses not already occupied, can provide a coherent overall experience with potential benefits over unimodal systems (Oviatt, 1999). In relation to pedestrian navigation, where the user’s visual and auditory perception is used to navigate the environment and pay attention to traffic, the sense of touch is rarely used. This presents an opportunity to explore how haptics might be used in navigation interfaces, to support the visual and auditory senses (Pielot et al., 2012; Rodríguez et al., 2019).

An Opportunity for Smartwatches

Another digital wearable utilizing haptic feedback that has seen increased adoption and popularity in recent years is smartwatches (Rimol, 2021). Compared to smartphones, smartwatches can contain the same required hardware to enable navigation services and haptic feedback, albeit with a smaller screen size for displaying map data. Where smartwatches stand out in terms of utilizing haptic technology is that they are worn directly on and in contact with the skin on the user's wrist at all times. Even if the skin on the wrist is not as sensitive as, for example, the hand and fingers, it still offers great tactile acuity for vibrotactile stimulation (Cholewiak & Collins, 2003) while still being unobstructed and keeping the hands free for other interactions (Heikkinen et al., 2009). The smartwatch presents an opportunity to hide haptic technology in a wearable that is already socially established, as well as placing it near an acceptable part of the body and still visually accessible to the user (Oakley et al., 2006). Unlike carrying a smartphone, which might not always be carried close to the skin, the smartwatch provides an always available source of feedback through haptics that does not require or conflict with the user's visual or listening attention. Since it can be worn and does not require the user to hold it, it provides the affordance of always being available, allowing the user have their hands free for other purposes (Rawassizadeh et al., 2014).

Smartwatches present an interesting opportunity to explore alternative approaches to pedestrian navigation supported by haptic technology, but these opportunities have not yet been embraced by the companies developing the standard navigation services available on these devices. The default navigation services that ship with most smartwatches are essentially an extension of the classic map-based approach that is the default on the smartphone they are paired with. Brands like Apple, Samsung, and Fitbit, use navigation services like Google Maps or Apple Maps to provide turn-by-turn navigation. The haptic interface used with these services is primarily to alert the user of when to make a turn and when the user has arrived at their destination. Taking the Apple Watch as an example, when the user is suggested to take a left turn to follow a predefined route, two haptic taps three times in a row means turn left, while twelve steady taps mean turn right, and a long vibration means that the user has arrived at the destination. At the start of this project, this interface was tested while navigating by foot, showing that the haptics alone was not sufficient to guide the user to a location that was unfamiliar to them. At the time of this writing, Apple has not disclosed if this is intended as an attempt to provide eyes-free navigation or just to support existing screen-based information, or potentially both.

Although the use of existing technology implies its own benefits and constraints to the design process, other devices embedded with haptic technology were also considered for this project, including the development of a custom device. However, developing a new piece of digital wearable for the sole purpose of improving pedestrian navigation raises the question of what added value such device can offer a potential user in contrast to the inconvenience of having to purchase, wear, and manage yet another digital device. Working within the constraints of designing on existing smartwatch devices — including the limitations of the available operating system and APIs provided to these devices — has the added benefit that any proposed design can be installed and tested directly on users existing choice of smartwatch and wristband material which they are already accustomed to and comfortable wearing, and which could lead to a more holistic examination of how users might wear or use the prototype in everyday scenarios.

Technical Challenges of Today's Navigation Technology

The technical challenges of designing a pedestrian navigation system include the need for high fidelity data about the user's position, direction, and data about their surroundings. The interface design challenge then lies in how this data is communicated back to the user in an appropriate, informative and convenient manner. Ideally, a perfect system would use accurate and up-to-date data about the user's position and direction, combined with absolute geographic and architectural data about the user's surroundings. The reality, however, is that although the accuracy of GPS systems in smartphones is (at the time of this writing) typically accurate to within a five meters radius around the user's actual position *under clear sky ideal circumstances*, it can decrease significantly depending on the height and material of nearby obstacles like buildings and terrain, and overhead objects like trees and roofs (Diggelen & Enge, 2015). Furthermore, the fidelity of today's map data — which can vary based on the service provider and location coverage — is less likely to include smaller walking paths, passages, alleys, etc., compared to larger objects like roads, streets, and highways. Although improving either of these — the precision of GPS or the fidelity of map data — is outside the scope of this project, a proposed system must still accommodate imperfections with currently available technology.

Related Projects

Several research projects have examined the use of wrist-worn haptic interfaces for spatial guidance purposes, utilizing the advantages of eyes-free HCI (Guo et al., 2009; Schätzle et al., 2006; Sergi et al., 2008; Weber et al., 2011). Many of these projects outline the benefits of using a wrist-worn device that is unobstructed and always in contact with the skin (Heikkinen et al., 2009), while at the same being time socially acceptable and does not occupy the hands for other interactions (Hong et al., 2016). Several projects have explored using the skin on the forearm using different types of wristbands and armbands (Hong et al., 2017; Ng & Man, 2004; S. Panëels et al., 2013; Sergi et al., 2008). Spatial directional sensing using tactile cues on the wrist has been shown to be able to successfully direct a users attention towards a target direction (Brodersen, 2019; Cammann et al., 2017; Jacob et al., 2011; van Erp, 2001; Weber et al., 2011), although also showcasing various proposals to how this information can be interfaced to the user. For pedestrian navigation purposes specifically, these proposals typically involve either:

- the use of haptic icons with embedded meanings associated with specific navigation instructions the user must follow to navigate to the target destination (Cammann et al., 2017; Ghouaiel et al., 2013), or;
- the use of one or more tactors placed strategically on the body to communicate the horizontal angle (bearing) towards the target destination relative to the user's current position (Heuten et al., 2008; Jacob et al., 2011; 2012; Jones, 2017), or;

- some combination of both approaches (Dobbelstein et al., 2016; Pielot et al., 2010b).

To differentiate the interactive approaches suggested by haptic researchers, Fröhlich et al. (2011) suggest two predominant types of haptic interface design exemplified using the following two metaphors:

The Magic Wand describes how a hand-held device can be used as a pointing device, directed at locations around the user. Using haptic feedback, this hand-held device can communicate information back to the user — through haptic vibrations in the device — about what it is being pointed at. For example, in a navigation context, a user might point in a direction of a road and get feedback about if this is still the correct direction to go. The magic wand approach can be made possible using GPS to get information about the user's geographical location, combined with a digital compass embedded in the device to get the horizontal angle that the user is pointing. Although The Magic Wand metaphor is suggested as a hand-held device, wearing the pointing device on the wrist would leave the hands free for other activities. Another consideration with the Magic Wand approach is the requirement that the user must be actively pointing in order to get relevant feedback. However, this can also serve as a way for the user to control when or at what intervals to receive the feedback.

The Sixth Sense metaphor describes how haptic information can be delivered to the user discreetly and effortlessly from a haptic display as if this information came to the user as a hidden sixth sense. Using this approach does not necessarily require the haptic display to be positioned at a specific location on the body, as long as it can be efficiently felt by the user and consistent with the direction the user is moving. In a navigation context, this approach could be used to discreetly signal information to the user as they are walking, if they need to make changes to their route or if they have arrived at their destination.

Both metaphors are interesting in that they suggest approaches to haptic interface design that serves related but different affordances. The Magic Wand approach suggests a way for the user to communicate with the device about their intentions and get immediate feedback relative to their embodied interaction. The Sixth Sense approach exemplifies how the use of haptics can be designed to be unobstructed, discreet, and private. This project explores the possibility of combining both metaphors in the final interaction design.

Methodology

The project presents a research approach to exploring potential futures of how pedestrian navigation can be improved using haptics and smartwatches. In doing so, various solutions and features of the design and development process are explored and evaluated. However, it is worth noting that the nature of the problem does not inherently imply an optimal or ideal solution due to the multitude of ways in which the problem can be framed and evaluated. The problems of pedestrian navigation involve an examination of the diversity of users' behavior and the variety of use cases and contexts in which potential solutions can be evaluated. Any solution will have to face the paradox of providing sufficient and timely information to the user while at the same time not distracting them from the reality of their surroundings. These challenge of pedestrian navigation in today's urban, suburban and rural environments presents a wicked problem. Potential solutions can be evaluated from a:

- **Commercial perspective**, evaluating how feasible the presented solution could be as a potential business case.
- **Safety perspective**, evaluating how using the proposed solutions can lead to safer pedestrian navigation in traffic under different circumstances.
- **UX perspective**, evaluating how solutions influence user behavior in terms of characteristics like usability, usefulness, and desirability.
- **Technical perspective**, evaluating how the solution might relate to other solutions in terms of navigation accuracy, power consumption, integration, etc.
- **Anthropological perspective**, evaluating how possible solutions can influence behavior, both alone and in a social context.

These perspectives are neither mutually exclusive or complete. They underline different ways of approaching the same problem and evaluate design outcomes. Although the focus of this project is to explore potential design ideas as an inquiry to doing research and answer the research question, all the listed perspectives present lenses through which the designer can see opportunities to extract meaning from doing design. For the purpose of exploring the unknown, a Research Through Design (RtD) approach was used, focusing on the design, development, and testing cycle as the primary means for generating knowledge (Zimmerman et al., 2007). This process of conducting RtD focuses on the role of the prototype, using it as an instrument of inquiry for generating design knowledge (Keyson & Bruns Alonso, 2009). The approach is useful to explore complex interactions and issues within a realistic user context relevant to the project’s scope. The designer can reflect back on the process and decisions made, not only based on the observed user interactions and experiences of the prototype but also through the designer’s own reflections working with the prototype. The experiential and often tacit knowledge obtained through this design activity is an essential part of the design practice (Schön, 1983). By observing the design process, observations can be used to guide further research as an iterative process and help focus the design space.

The creative process was followed by taking notes throughout the project, documenting relevant findings, which, together with the developed prototype and this report, comprises the research outcome. The project was supported by examining relevant and related projects to draw inspiration and knowledge from existing work in the areas of haptic HCI, wrist-worn haptic interfaces, and pedestrian navigation (see Related Work).

Testing

As part of the iterative design process, participants were used in pilot tests, preliminary tests, and to provide feedback on the prototype at various stages of development. Additional participants were invited using a public link and email invitations to onboard additional testers using the beta testing service TestFlight[1].

Tester	Sex	Age	Smartwatch user
#1	Female	26	Yes
#2	Female	27	No
#3	Female	28	Yes
#4	Male	30	Yes
#5	Female	61	No
#6	Male	64	No

Table 1: A total of six testers participated in in-person testing of the prototype at various stages of development. Three of these were already smartwatch users.

A total number of six participated in the in-person testings (see Table 1). During in-person testing, participants that were not Apple Watch users would borrow an Apple Watch with an adjustable wrist band. The participants were observed while using the prototype in different navigation situations in public, performing various tasks. The typical tasks involved getting familiar with the interface by navigating to a destination in full disclosure of where the destination location was and also allowed them to look at the user interface to learn the association between the visual and haptic feedback as they approached the location. Gradually the participants were exposed to more complex tasks, such as navigating to a destination that was secretly chosen, to the most difficult; using the app without looking at the user interface and by relying on haptic feedback alone (the UI on the Apple Watch was hidden programmatically to avoid accidental peeking). All testing of the prototype was done outdoor, in public, and in urban or suburban environments. The participants were questioned using an unstructured interview approach both during and after navigation about their experiences. Unstructured interviews were used to ask about the participant's experiences as they happened and allowed to adapt the questions according to the events of the navigation.

Some participants were only used for testing once at the end of the project, while others participated in several trials throughout the project to test changes in the design. During testing, the participant's ability to navigate to a location, the observer was walking along in the background, using the paired iPhone to monitor the status of the testers Apple Watch, the system's interpretation of the user's location and orientation, and in relation to waypoints and the final destination on a map only visible to the

[1] TestFlight https://developer.apple.com/testflight/

observer (see Figure 1). Depending on the task and focus of each test, the testers were questioned about different aspects of the system and what they were experiencing. To test alternative behaviors of the system interface, the observer was able to perform changes to the interface remotely on the iPhone and get feedback from the tester about the impact of these changes.

During testing, navigation data were also captured, including timestamps with route and location data, GPS accuracy, changes in motion gestures, and activity type at different intervals, from the start and until the goal was reached. This data made it possible to revisit a navigation event, find bugs and issues with the prototype or hardware sensor's accuracy, and understand the participant's behavior at key points in their navigation.

Figure 1: Test setup. The observer is using the smartphone app to view what the participant is experiencing and information about the ongoing navigation.

Design

The approach used in this project is inspired by the use of a compass as the primary mediator of implicit directional guidance on the smartwatch. This section describes the main challenges and considerations that went through the design of a multimodal smartwatch interface and an interaction model to accommodate both visual and haptic "eyes-free" navigation. The proposed interaction design is exemplified in the multimodal interface used in the final project prototype.

Using a Digital Compass for Directional Guidance

Similar to a magnetic compass always pointing towards magnetic north, the digital compass suggested for the final prototype always points in the direction that the user must go to reach their chosen destination. In the app, this direction is displayed visually using a rotating ↗ arrow and haptic vibrations associated with these rotations to indicate if the user is pointing towards a chosen destination.

Navigating using a digital compass has a number of relevant differences compared to turn-by-turn navigation:

Learning Curve: The use of a pointing direction to indicate the direction to go is similar to the innate ability of humans to express spatial direction using the hand gesture to point. This familiarity of directional pointing as a mental model for how the interface works can result in a lower learning curve for new users.

Relatable: Knowing the direction (i.e., the *beeline*) towards a location means that the user can get a sense of their own presence relative to that location. This is in contrast to getting an explicit navigation instruction (like "turn left") that does not necessarily tell the user anything about where they are relative to where they are going. Both types of information can be supported by displaying the remaining distance that the user is from the final destination, giving more context to the user.

Exploration: By not being constrained to follow a specific route and turn-by-turn navigation, the user can feel free to explore their surroundings and choose their own way towards the destination (Dobbelstein et al., 2016). However, attempting to take the most direct route is not always the fastest approach when large obstacles like water, buildings, railroads, or impassable terrain are blocking the path. In these situations, the user can be better off being guided, following a route around the obstacles.

Data type: The direction that the compass is pointing can be expressed as an *angle* of how much the user must turn in order to face the direction of the compass, i.e., when the angle is zero, the user is facing in the same direction as the direction towards the target. The value of this angle is relatively easy to visualize in a haptic display, either using haptic icons to indicate a range of intervals (as seen in Pielot et al., 2010a), or by changes in rhythm or amplitude based on how close or far the angle is from the zero-angle (as seen in Pielot et al., 2011b). In cases where a navigating user encounters multiple route options not marked with visible street signs, these options can be challenging to describe linguistically without more detailed mapping data. In these cases, the use of a pointing direction towards the route to take can be faster and more convenient than having to look around for a street sign. On the contrary, confirming one's location using street signs when they are available can be reassuring and help prevent navigation errors.

Accuracy: As the user moves away from the beeline to the target destination (such as to walk around a blocking object), the compass will have to update to reflect the new angle towards the target. How much this angle change is directly related to the remaining distance from the user to a target: If the target is very far away, this change in angle is minimal. For example, if a user is standing some distance away from a target and then walks 50 meters to the left or right away from the beeline (still maintaining the same relative distance to the target), then the angle to the target will naturally have changed. If this distance is 1 kilometer, the user's compass will only have changed by 2.87°; if instead the distance is 100 meters, the change is 28.96°. Using directional navigation means that the precision of the GPS signal relative to the precision of the direction displayed to the user is less of a concern at a distance than when the user gets close to the target. If the user is far away from the target and can get a sense of the general direction to the target using the arrow, this benefit can mean that the user is less inclined to check their device for changes repeatedly. This is, of course, only relevant unless waypoints are used, i.e., how many and how they are placed on a suggested route. If a waypoint is used on every turn where turn-by-turn navigation would also be used, then this advantage is diminished.

Determining the Direction of the User

With car navigation, determining the direction of the car is relatively simple. It is the direction that the car is facing, potentially supported by a compass attached to the frame of the vehicle. However, for a pedestrian using a smartwatch with a compass, the only direction known to the system is the direction of the smartwatch. Since a user is able to move their arm and hand freely, this direction of the smartwatch can be in potentially any position and not necessarily in the same direction as the user is facing. This problem of relating directional haptic feedback to the user's navigation direction has led researchers to explore solutions such as embedding the navigation device into belts (Pielot et al., 2008; Tsukada & Yasumura, 2004), around the torso (Van Erp, 2005), or head-mounted (Kerdegari et al., 2016). These locations are relatively stable relative to the body-facing positions and stable on the horizontal plane used when guiding to geographic locations. Solutions to the challenge of expressing a direction on the wrist regardless of arm/hand orientation has been done previously (Brodersen, 2019; Sabrina Panëels et al., 2013), although with the use of multiple tactors around the wrist for the added dimensionality that it provides. Since most currently available smartwatches contain only one haptic engine placed on top of the wrist, using multiple tactors was not considered for this project and presented an interesting design challenge.

The approach suggested in this project is to track the motion of the smartwatch and use the Magic Wand metaphor by detecting when the smartwatch (and thereby the hand) is held in a position where it can be used to point in a horizontal angle. As with a compass that is required to be held horizontally to work, the same constrain is used in the prototype interface design. In order to point in a horizontal direction, however, it requires that the user actively hold their hand in a position where the direction of the compass makes sense (where the screen is horizontal). Although this approach seems to only limit the usability of the design at first, the gesture of moving the hand up to horizontal or down again can be used as a toggle switch for the user, such as to control when to enable and disable directional feedback and haptic vibrations:

- When the user **raises** their smartwatch — such as to glance at the clock to read the time — the same gesture can be used to check the direction of the active navigation by enabling feedback (see Figure 2).
- When the user **lowers** their smartwatch again — such as with the hands down alongside the body, when walking or standing — this stance can function as a gesture to pause the directional feedback.

Regardless of the arm orientation, the user still continues to receive notifications about when the destination or a waypoint has been reached, akin to the Sixth Sense metaphor. In other words, the user does not have to keep their hand raised to navigate — only to check on their current direction relative to the next waypoint or the final destination.

With the ability to control the interface with a gesture, the user can control when and for how long to be exposed to vibrations. Although a different approach with continuous feedback would mean that the user would be able to feel the direction constantly throughout the duration of the navigation, this could lead to tactile stimulation fatigue, desensitization, and discomfort (Craig, 1993), as well as battery drain on the smartwatch.

Figure 2: The arrow is always pointing towards the destination location. As the user rotates their smartwatch, the direction of the location changes. The green arrow indicates that the user is pointing their watch at the target destination.

Using Haptics for Directional Guidance

With the hand raised to view the compass arrow, another challenge in the design is how to relate the watch's horizontal direction with the user's direction. In other words, how does the device know in which direction the user is facing in order to provide the relevant haptic feedback. Visually, the compass arrow can suffice the user's need by showing graphically the direction of where to go — regardless of the user's own facing direction. However, to indicate using haptics how much the user must turn to face this direction, i.e., the *bearing* (horizontal angle), requires that the device can relate the direction of the arrow to the direction that the user is facing. At the start of the project, the initial design used the pointing direction of the hand that the smartwatch is worn on, similar to the Magic Wand metaphor, although with the "wand" attached to the wrist instead of being held in hand. To use this approach means to point in the direction of the hand on which the smartwatch is worn as if pointing with the hand. Specifically, this is using the right side of the screen as the pointing direction if the hand is worn on the left hand. However, during testing, it was discovered that this posture was not natural for the participants and was found to be awkward to perform in public. An alternative approach was also proposed, to instead uses the top of the smartwatch screen to point. Using the top of the screen means that the smartwatch can be held up in front of the chest, using the same gesture as when a person checks the clock (see Figure 3). The top of the watch is then pointing in the same direction that the user is facing. This approach became the most popular option amongst testers (see Findings).

With knowledge about the direction that the user is facing and the direction towards the next target, the bearing can be calculated. A bearing of 0° means that the user is facing perfectly in the direction of the target. A bearing of 180° means that the user is facing directly away from the target. This change in bearing — although simple at first — can be expressed in numerous ways, several of which were tested through the design and development phase:

As the user is turning towards the target, the bearing approaches 0°. Using haptic feedback to express this gradual change in bearing to the user can be done by changing the frequency and interval between haptic vibrations. As the user closes in on the target direction and the bearing approaches 0°, the haptic frequency increases. This approach creates familiarity as it is similar to the proximity alarm in some car parking systems that also beeps with increasing frequency to warn the driver when the car gets closer to nearby obstacles. The advantage of the gradual approach is that even if the user is not pointing exactly at the target, the feedback can still give the user clues about how close or far off the user is while the user continues to locate the target direction. However, this haptic signal is also more complex, as the user would have to get used to the association between the bearing and the change in frequency. Another drawback is that a user unfamiliar with the limit of the increase in haptic frequency might overstep their rotation until they start to feel a decrease in frequency again. Although this interaction can be improved with practice, it requires mental memory about the range of frequencies used by the app.

When testing this approach, it was realized that this approach required considerable focus if the starting direction was off, as the user had to focus their attention on the haptic interface, potentially diminishing their attention to their surroundings. Focusing the attention on the haptic interface also meant that the participant would stop walking. To check the direction, the user would rotate slowly to feel how their change in direction impacted the feedback they received. This 'rotate → feel → rotate → feel'-activity loop was considered inefficient in practice. If trying to feel the direction took too long, the participants would just look at the smartwatch screen directional arrow instead for a quick confirmation, questioning the affordance of using a haptic interface altogether.

Another approach tested was to use a simple binary signal: If the user is pointing towards the target, the user feels vibrations; if the user is pointing away, the user feels no vibrations. The drawback of not knowing how close or far away the user is from the target (and in which direction to rotate to get there the fastest) was less of a problem than anticipated. Using the binary approach meant that the interface could only be in two states, which makes it very easy to react to. The user can rotate more quickly than with the gradually increasing haptic and react to the haptic feedback when they are pointing in the right direction. Using this approach means that a tolerance angle must be specified for when the user is considered pointing towards the target. If this angle is zero or too narrow, the user has to point almost exactly at the target, which is both difficult to do and impractical for getting quick confirmation of the current direction. If the angle is too wide, the user might become unsure about the actual direction of the target and make wrong decisions on what route to take. Using a too wide angle also means that the user has to check the boundaries of this angle to estimate a midpoint as the direction towards the target. Other projects have tackled this problem of specifying an ideal bearing tolerance, suggesting that no perfect solution exists, as different people and different contexts can call for different precision in directional sensing (Dobbelstein et al., 2016), with one suggestion that a dynamic angle could be used based on for example the number of routes options available at any given location (Robinson et al., 2010). In the final prototype, a default horizontal tolerance angle of 30° was used (15° to either side of the target) and with the option to change this value in the app's settings.

The final prototype presents two approaches to how the haptic interface is used for directional guidance. Both types use the same style of continuous haptic vibration when the user is pointing towards the target. Where they differ is how they act when pointing away from the target:

- **Proximity:** When pointing away from the target, the smartwatch vibrates using a haptic pattern with intermediate delays between them. The interval of these delays increases based on how far away the user is from pointing at the target. This is similar to the gradual approach explained earlier, but with a unique haptic vibration when pointing at the target (within the threshold) to avoid the issue of overshooting.
- **Compass:** When pointing away from the target, the smartwatch only vibrates subtly with a mechanical tick vibration every 15° of rotation. This approach is the simplest and resembles the binary approach discussed earlier, only with the added tick. This tick indicator was added only to sense the rotation on the smartwatch and to give feedback to the user that the app was monitoring their rotation.

Figure 3: Two different pointing settings for the smartwatch. The left photo shows the user is pointing out from the *top* of the watch screen, facing away from the user. The right photo shows the user is pointing out from the *right side* of the watch screen.

Although some testers liked the idea of Proximity, in the end, it was the Compass approach that was preferred by all testers for its simplicity. Attempts were also made to improve the interface by using distinct haptic patterns for *left* and *right* directions to make it possible to feel to which direction from the target that the user was pointing, to make it easier to turn in the right direction. However, it was observed that this addition made the interface too complex.

Using Haptics for Distance Sensing

Alongside the direction to a target location, the distance to the target is also relevant for the user to help them orient themselves and to detect progress as they navigate. Visually, this is presented numerically on the smartwatch as the distance to the destination (optionally also to the next waypoint). Three options for indicating distance using haptics were explored, including the option to disable it:

- **Proximity:** When pointing at the target location, the user receives continuous vibrations. With this option, the frequency of these vibrations becomes faster as the user approaches the target. Instead of using a linear increase in frequency, the increase was done using three discreet tiers to make the change in the distance more noticeable. These tiers only kick in and increase the frequency when the user gets sufficiently close to the target. The tiers are more than 160 meters, between 160 and 80 meters, and less than 80 meters. Less than 40 meters means that the target has been reached. Proximity is, therefore, only useful to get a sense of approaching the target; not to read the exact distance.
- **Intervals:** When pointing at the target location, a vibration is felt for every 100 meters remaining, followed by a pause. Then, the pattern repeats itself, i.e., 1200 meters is felt as 12 vibrations followed by a pause. This approach is more precise than proximity when the distance is above 100 meters but also more demanding, as the user would have to count vibrations, making it impractical for longer distances.
- **None:** This feature is disabled. The distance can only be interpreted visually.

Of the first two options, these do not express the exact distance in meters but only rounded to the closest 100 meters (Intervals) or a sense of getting close (Proximity). Trying to express the exact distance down to the meter would require a more advanced interface, potentially introducing more complexity than what is beneficial to the user. This should be contrasted with the user’s option of glancing at the smartwatch screen for visual confirmation. For most testers, though, this feature of feeling the distance was deemed too complex when first learning the haptic interface, although the idea of feeling the proximity was appealing.

Creating Eyes-Free Navigation

Using the described interaction design patterns, the prototype presents a multimodal interface providing directional and distance information feedback to the user, both visually and using haptics. This enables the user to take advantage of both or either sensory modalities depending on the use context and what is most convenient to the situation. If the user is navigating and focused on paying attention to their surroundings, they can resort to simply raising the smartwatch briefly to engage with the haptic interface. This way, they can get quick confirmation about the direction they are going or how they need to change their direction — all without looking at the screen. When the user reaches their destination, they will feel haptic vibrations indicate that their navigation has finished. If the user is using waypoint navigation, they will feel a different vibration pattern indicating that the direction has moved to the next waypoint. They must then raise the watch to scan for the next waypoint location and continue the walk in that direction. If the user is interested in more detailed information about the current navigation, they can always confirm the state of the navigation visually on the smartwatch's screen. Even when the eyes-free aspect of the proposed design is the focus of the navigation, the UI plays an important role in associating the haptic feedback with visual and precise information to create an appropriate mental model of how the navigation and interface works.

Although the user is able to use either visual, haptic, or both feedback modalities in the final prototype, the feature of being able to hide the UI on the smartwatch during testing was added for testing purposes only (see Figure 4). With practice, users who had gained experience with the navigation style and the haptic interface were able to locate a secretly selected location and navigate to it using haptics alone (see Findings).

Figure 4: On the left, the arrow is visible. On the right, the arrow is hidden.

The Final Prototype: Haptic Navigator

This section presents the final version of a working pedestrian navigation tool named *Haptic Navigator* (see Figure 5). The presented prototype consists of both a smartwatch app for Apple Watch and a companion smartphone app running on the iPhone paired with the Apple Watch.

Although the smartwatch app can be used without launching the smartphone app, the smartphone app can be useful to make certain actions more convenient, like typing on a keyboard to search for a destination or viewing multiple or complex types of information. Starting and stopping the navigation can also be done from both the smartphone and smartwatch apps, but the activity of navigating, and sensing motion and direction, happens through the smartwatch. The user can rely solely on using the smartwatch app if preferred, but utilizing the smartphone in this project provided several benefits which are elaborated later. Although the app prototype was developed and tested on Apple Watch Series 5 and newer devices, the presented design is not restricted to Apple hardware and software. The app concept could essentially be ported over to a similar solution for Google Wear OS or other suitable smartwatch platforms.

A Tool for Wayfinding

To describe how the app works as a tool for navigation, we first address how it can be used for its primary purpose of pedestrian wayfinding. Typical wayfinding consists of a four-step process involving the following steps; *Orientation*, *Route Selection*, *Route Control*, and *Recognition of Destination* (Downs & Stea, 1973).

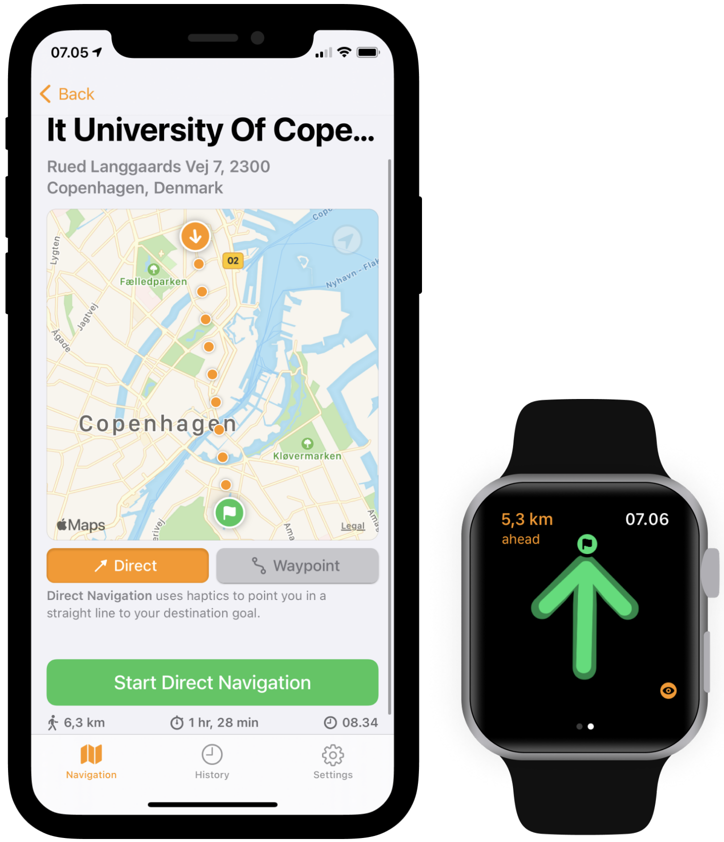


Figure 5: The final prototype called Haptic Navigator.

Orientation

At the start of the wayfinding process, the user must orient themselves and find out where they are with respect to their surroundings and the destination they wish to navigate to. This step is highly influenced by the users existing knowledge about their surroundings and the destination. When using a navigation device, orientation implies that the device is informed about where the user is and where the user wants to go.

Haptic Navigator presents the following user flow to achieve these steps:

- To start a new navigation, the user must first select where they want to navigate to (see Figure 6). This can be done in one of two ways:
    - **Search for a destination using a name or address.** As the user enters a query in a search field in the app, the app suggests a list of destinations that matches the query. To make the suggested destinations more likely to be relevant, the user's current location in the world is taken into account so that search results closest to the user are shown at the top.
    - **Search for a destination using a map.** The user is presented with a map around the user's current location and can then pan around the map to place a map pin ( ) at any location. As the pin is placed, the app informs the name or address of the location where the pin is placed. This then has to be confirmed by the user as the destination to use.
- After choosing a destination, the user is presented with a pre-navigation screen, where the user can confirm information about the location and route and start the navigation (see Figure 7). This screen contains the following information:

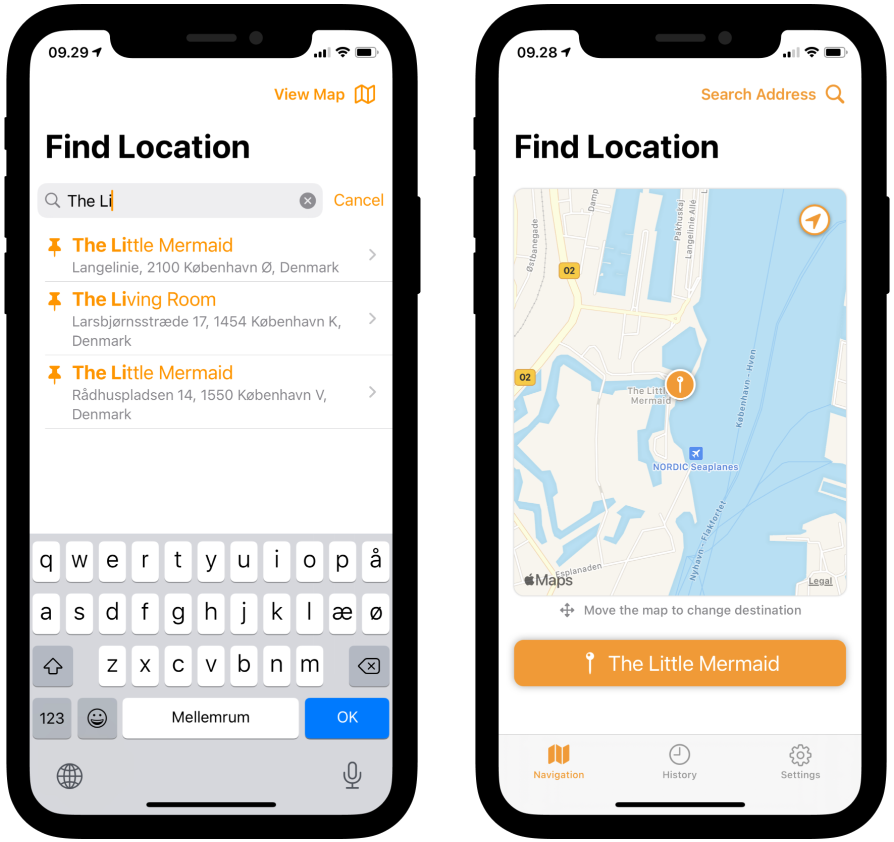


Figure 6: On the left, searching for a destination by typing the name or address of the location. On the right, searching for a destination by placing a -pin on the map.

- A map where the user can see their current location and the location of their chosen destination.
- An option to switch choice of *navigation guide* (see below) and view the suggested route based on this choice, before starting the navigation.
- Meta information about the navigation, including:
    - Estimated travel distance (using a walking path)
    - Estimated travel duration (by walking)
    - Estimated time of arrival (by walking)

Although this flow of searching and selecting a destination, confirming the navigation information, and starting a navigation can be done from both the smartwatch and paired smartphone. The smartphone is generally easier to use and can be put away again when the navigation has been started on the smartwatch.

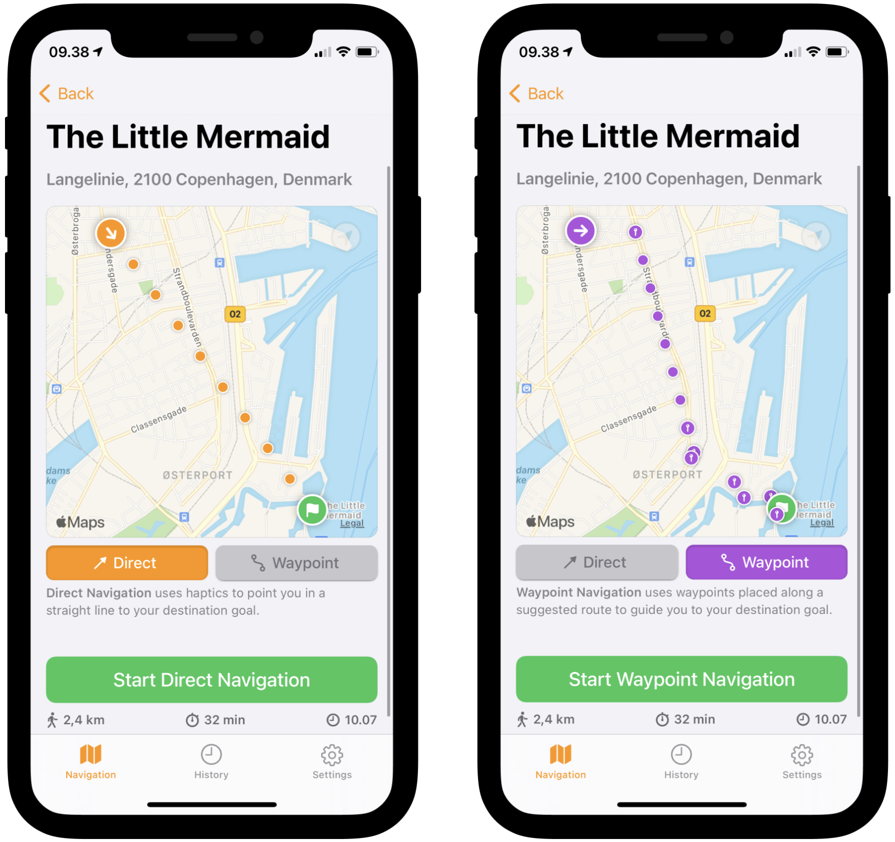


Figure 7: Pre-navigation screens showing a map with the two choices of guidance: *direct navigation* or *waypoint navigation*.

Route Selection

A user walking to a destination typically has the choice between several routes to take. The choice of route can vary depending on a users' subjective priorities such as:

- Taking the (perceived) fastest route or most direct route.
- Taking the most convenient route, mentally (e.g. using familiar or well-defined roads) or physically (e.g. avoiding obstacles, steep or bumpy roads, heavy traffic, etc.).
- Taking the most enjoyable route (e.g. sightseeing, exploration, etc.).
- Any combination based on personal preference.

The app presents two ways of being guided to the final destination, which controls in which direction the navigation arrow should point. The choices are:

↗ **Direct navigation.** The arrow always points directly at the final destination, leaving it up to the user to decide which route to take to get there.

Waypoint navigation. The arrow always points at the next closest waypoint along a suggested path to the final destination. As the user approaches a waypoint (or walks a sufficient distance past it), the waypoint is cleared, and the arrow switch to point at the next waypoint on the route until the final destination is reached.

Before starting a navigation, the user can see all waypoints forming a suggested route from their location to the chosen destination. When a navigation is started and in progress, however, the smartwatch only displays the direction to the next waypoint using the compass arrow.

For more information about the ongoing navigation in progress, the user can use the smartphone to view a map of all relevant information, including the user's current location, the destination location, and the location of all remaining waypoints, including which one the smartwatch is pointing to next.

The user is always able to switch freely between direct navigation and waypoint navigation at any time. If direct navigation is chosen, the system still clears waypoints in the background without notifying the user. This is done to keep the state of the waypoints relevant if the user suddenly decides to switch to waypoint navigation.

Route Control

When a user is trying to find their way to a destination or intermediate location, it is important to get up-to-date information about the state and progress of their navigation.

In Haptic Navigator, the information provided on the smartwatch UI is:

- The compass ↗ arrow;
- The remaining distance to the final destination;
- Feedback about when the destination has been reached.

If waypoint navigation is being used, this information also includes the distance to the next waypoint (if any), as well as visual and haptic feedback whenever a waypoint has been cleared, so the user knows to check the direction again for the next waypoint. When the final destination has been reached, the navigation is finished automatically.

From both the smartwatch and smartphone, the user is able to control the following actions during an active navigation (see Figure 8):

- Pause/Resume the navigation
- Finish the navigation prematurely
- Disable/Enable haptics
- Change navigation guide (between direct and waypoint navigation)

Furthermore, the smartphone can also be used to change and experiment with settings (in the app's Settings screen), which will take effect on the smartwatch immediately without interrupting the navigation in progress.

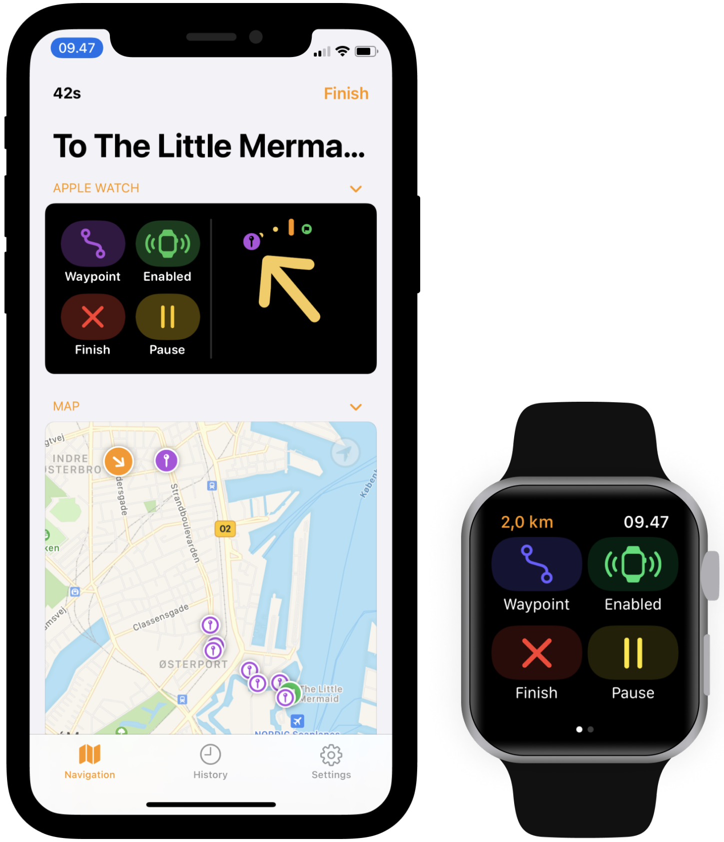


Figure 8: The user can control the navigation from both the smartphone and smartwatch.

Recognition of Destination

The user might be unfamiliar with how their destination looks and, therefore, might risk walking past it. Thus, it is important to inform the user — preferably ahead of time — that they have reached their destination or are about to do so. In Haptic Navigator, a user has reached the final destination if their geo-location is ever measured to be within a radius of 40 meters around the destination coordinates, or between 25–40 meters for waypoints, depending on current GPS conditions. Although this choice of geofence radius was chosen based on experience testing the app, a once-size-fits-all geofence radius is not always ideal (see Findings).

When reaching the final destination, the Haptic Navigator presents a "Finished navigation" screen followed by haptic feedback to inform the user that they have arrived at their destination (see Figure 9). On the finished navigation screen, the user can also see more information about the navigation that has just finished, such as the estimated traveled distance, duration, etc. This screen is also accessible if the navigation is finished manually before the destination has been reached, although without displaying the "Destination Reached" animation.

Figure 9: Screens shown when the destination is reached.

Sensing Direction as a Reaction to Interaction

Using the smartwatch's sensors to track the user's location, pointing direction and arm/wrist motion, then combine the data to compute the appropriate feedback and deliver it back to the user in a timely manner, is requires to provide a responsive and *fluid* interaction between the user and the interface. The user can receive feedback both *passively* as they walk or stand still and notices the feedback relative to their current location; and *actively* as they explore their surroundings by pointing and receive instantaneous feedback from their movement. Although wearable tactile devices, like the smartwatch using vibrations, are categorized as using cutaneous passive haptics (Rodríguez et al., 2019), the instantaneous feedback loop from detecting wrist movement and providing tactile stimulation accordingly, as if the user was actively hitting some object with their wrist, enables and supports an active exploration of the interface (see Figure 10).

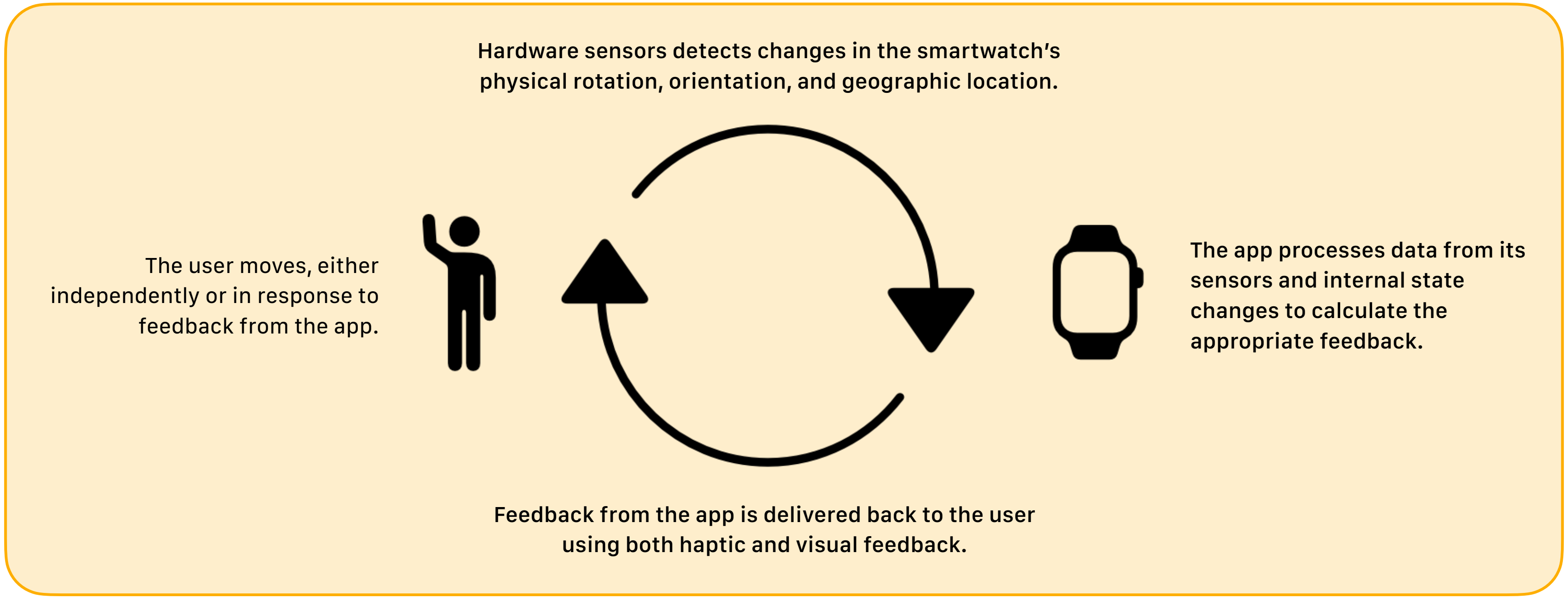


Figure 10: The information flow between the user and the smartwatch.

Taking Advantage of the Smartphone

Most smartwatches do not live in a vacuum; they are often used as an extension to a smartphone to present information and to exchange data and services. If one device lacks certain capabilities, then a common strategy is to take advantage of hardware features and capabilities that the other device can provide. For example, the smartwatch used in this project does not have cellular internet connectivity on its own. However, it can use a short-range Bluetooth connection to use the paired smartphone as a relay for internet communication. This enabled searching for a destination location on the smartwatch; by sending the request to the smartphone, which then forwards the request to an online map service; the service responds with search results which the smartphone delivers back to the smartwatch. Smartwatches usually have smaller screens, limited storage, and less processing power than the smartphone they are paired with, which means that some user activities like typing on the screen or viewing complex data like geolocations on a map are best done on the smartphone when it is available. This project is an example of this. Although the navigation experience could be conceived entirely on a smartwatch alone, it was realized during the project that creating a counterpart iPhone app for iPhone made working with the Apple Watch significantly more convenient and efficient. Four key advantages were behind this decision:

① Searching for and looking up locations to start a new navigation is considerably easier on the iPhone due to the larger screen and keyboard. Specific native map services were also only available on iOS and not WatchOS, such as searching for a location using a string query or searching for route suggestions.

② During testing on participants, the iPhone could be used as a remote monitoring device to:

- Remote control the Apple Watch to assist the tester at a social distance, such as to easily pause/resume or turn off vibrations during questioning.
- View additional navigation statistics for the currently active navigation, such as:
    - A map view of the tester's location (as interpreted by the GPS) and pointing direction of the watch (bearing) on a map.
    - Live GPS accuracy.
    - Waypoint's locations on the map.
    - Destination location on the map.
    - Metadata, including traveled distance, duration, remaining waypoints, etc.
    - Remotely viewing the direction arrow displayed on the smartwatch, including when the arrow was otherwise hidden from the tester for testing purposes.

③ Adjusting and fine-tuning advanced setting remotely on the smartwatch during active navigation.

④ View recorded navigation history data, such as timestamps recorded at time and distance intervals, and timestamps captured whenever an activity like clearing a waypoint, reaching the destination, changing navigation mode, or raising the hand to sense direction was recorded (see Figure 11).

Another advantage worth mentioning is how although the Apple Watch has its own GPS module, in cases where the GPS signal is poor, the prototype can attempt to use the iPhone's GPS signal to improve accuracy (in case the iPhone can receive a better signal). This was done by requesting location updates from the iPhone to the Apple Watch if needed and only using the returned data if it was sufficiently accurate or if previous readings were significantly outdated (older than 10 seconds). Due to the potentially limited space on an Apple Watch, the iPhone was also used to store and process navigation records (data collected throughout during an active navigation) to avoid using the limited space available on the Apple Watch.

Smartwatches are typically used as an extension to one's smartphone, providing quick access to glanceable information and simple interactions. This paradigm is also apparent in Apple's documentation, where only recently creating Independent Watch Apps has become possible[2]. These roles are switching in this project; the smartwatch is responsible for the navigation experience, and the smartphone is supporting that experience.

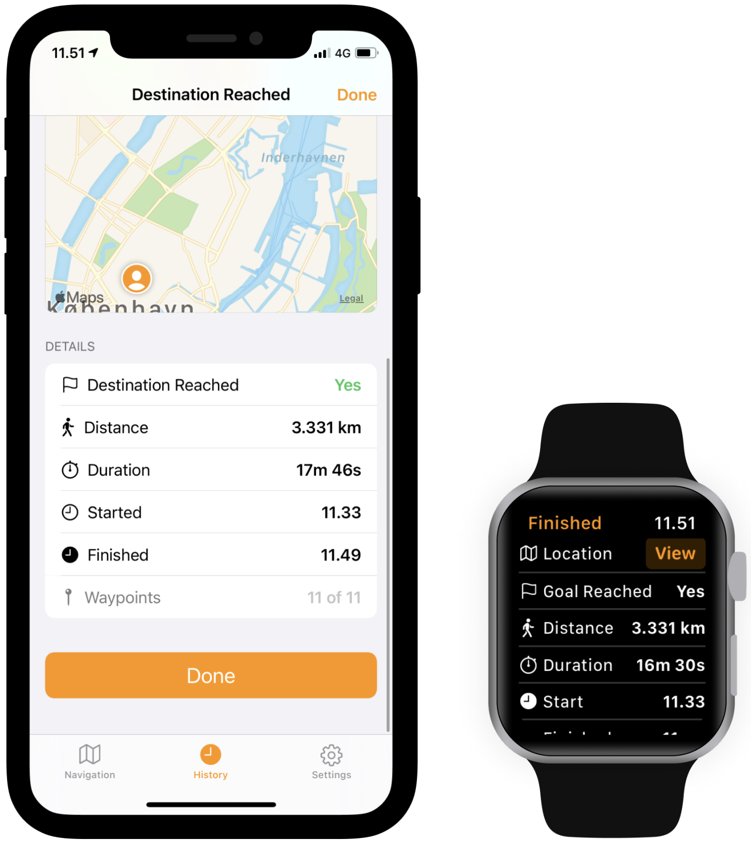


Figure 11: Navigation history data on smartphone and smartwatch. The data is presented when reaching the destination but can also be accessed later through the history section in the smartphone app.

[2] WWDC 2019: *Creating Independent Watch Apps* https://developer.apple.com/videos/play/wwdc2019/208/

Development

This session discusses some of the decisions behind the choice of hardware and software platform used in the project, including some of the challenges introduced by these choices. The development process is explained briefly, including an outline of how the system was designed in software.

Hardware Considerations

The Apple Watch represents a common modern smartwatch that includes the capabilities you can expect from typical consumer smartwatches available today. These capabilities include being a small wrist-worn mobile device with the following hardware components that were relevant for this project: A screen, a processor, a battery, Bluetooth or other capable wireless connectivity, a haptic feedback engine, GPS, a gyroscope, an accelerometer, a magnetometer, and an adjustable wristband to accommodate different wrist sizes.

Alternative hardware choices and platforms were also considered at the start of this project. Still, the choice of using an Apple Watch and iPhone was favored due to availability and familiarity, the capabilities of the hardware, and an assessment of the development ecosystem and available frameworks. Regardless of the choice of platform, it was considered that any choice would include a significant learning curve and potential platform-specific, vendor-specific, and hardware-specific constraints. For this reason, a Do-It-Yourself (DYI) approach was also considered, working with custom-built prototypes and sketching in hardware using hobby electronics, similar to previous projects (Brodersen, 2019). However, the benefits of working with existing technology and on devices people already own and are accustomed to have its obvious benefits (as discussed in An Opportunity for Smartwatches). A DYI approach can also introduce significant overhead and resource costs of acquiring and combining capable hardware components in a complex system and in a design that can be worn as a wearable on the wrist and be worn in public without being socially awkward testers.

Software Development

Developing the software on both the iOS and WatchOS platforms was done using the Xcode integrated development environment (IDE) and programmed entirely in Swift. GitHub was used for version control and TestFlight was used for distributing beta versions to testers. The project source code behind the Haptic Navigator app bundle consists of three code bases: One for iOS (iPhone app), one for WatchOS (Apple Watch app), and one for code shared by both platforms.

Although the project relied heavily on test-driven development and establishing a quick design and development loop, the use of haptic feedback, motion tracking, location tracking, and cross-platform communication presents a challenge. Typical IDE tools such as simulators and emulators are not sufficient to test non-screen-based interactions in a non-static context, such as testing haptics with motion-sensing while walking. Testing must instead be done on real devices and occasionally require the developer to move around and go for a walk to try a specific navigation feature and get a sense of distance, direction, and freedom of movement. However, while away from the IDE, the developer loses the benefit of diagnostic tools and developer consoles, as the devices must be disconnected from the IDE. For these reasons, several intermediary tools and debug screens were developed to give feedback about the state of the devices during testing, as well as to make sense of sensor readings in different contexts (see Figures 13 and 14).

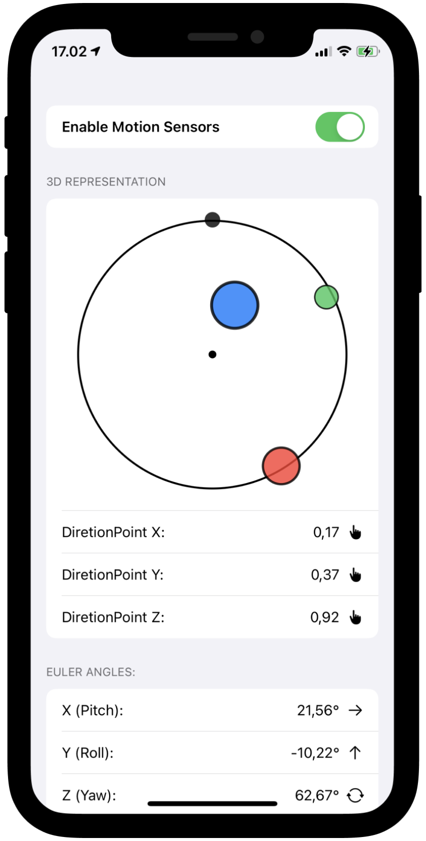


Figure 13: An early version of a debug tool used to experiment with, and debug, motion data. The red, blue and green circles indicate X, Y, and Z vectors in 3D space.

Figure 14: An early versions of the app and implementation of the directional guidance.

As the development progressed and the prototype interaction design became more refined, experimentation with various design modifications became a more significant part of the process. Some of these design decisions showed themselves to be more opinionated and subjective, advocating for the ability to make on-the-spot changes and tweaks directly on the device while testing for quick experimentation. This was especially true when working with participants and exploring alternative designs, by being able to modify the experience of the device remotely from the smartphone, affecting the experience by the participant on the smartwatch (see The Final Prototype as a Research Artifact and End-User Product). The final version of the prototype contains the following options in the Settings screen (see Figure 15):

- Select which arm (left or right) the watch is worn.
- Select preferred device pointing direction (Left, Right, Top, or Bottom), see Determining the Direction of the User.
- Set the default Navigation Guide (Direct or Waypoint).
- Select the type of haptic feedback style (Compass or Proximity). See Using Haptics for Directional Guidance.
- Select the specific type of haptic patterns to use for each feature of the haptic interface relative to the users pointing direction and the status of the navigation:
    - **On target**
    - **Left of target** (Proximity only)
    - **Right of target** (Proximity only)
    - **Compass tick** (Compass only)
    - **Waypoint cleared**
    - **Destination reached**
- Select the horizontal angle tolerance of how close the user must point towards the target to count as pointing at the target. Select a value between 1° and 45° (default is 15°).
- Select the vertical angle tolerance of how much the user must raise or lower their hand to activate or deactivate the pointing arrow and haptic feedback. Select a value between 1° and 45° (default is 20°).
- Select the style of haptic feedback to receive as indications of distance to the next target when pointing at the target (None, Proximity, and Interval). See Using Haptics for Distance Sensing.
- Toggle showing or hiding the visual animation of the arrow vibrating whenever haptics is vibrating.
- Toggle if the GPS algorithm used should attempt to place the user's location on nearby roads on the map (used in car navigation), or if the app should use raw location data to place the user independently of the map or other sensors.
- Toggle to save finished walks to count towards the user's activity goals (Apples Workout features).
- Delete all saved navigation data.
- Restore settings to defaults.

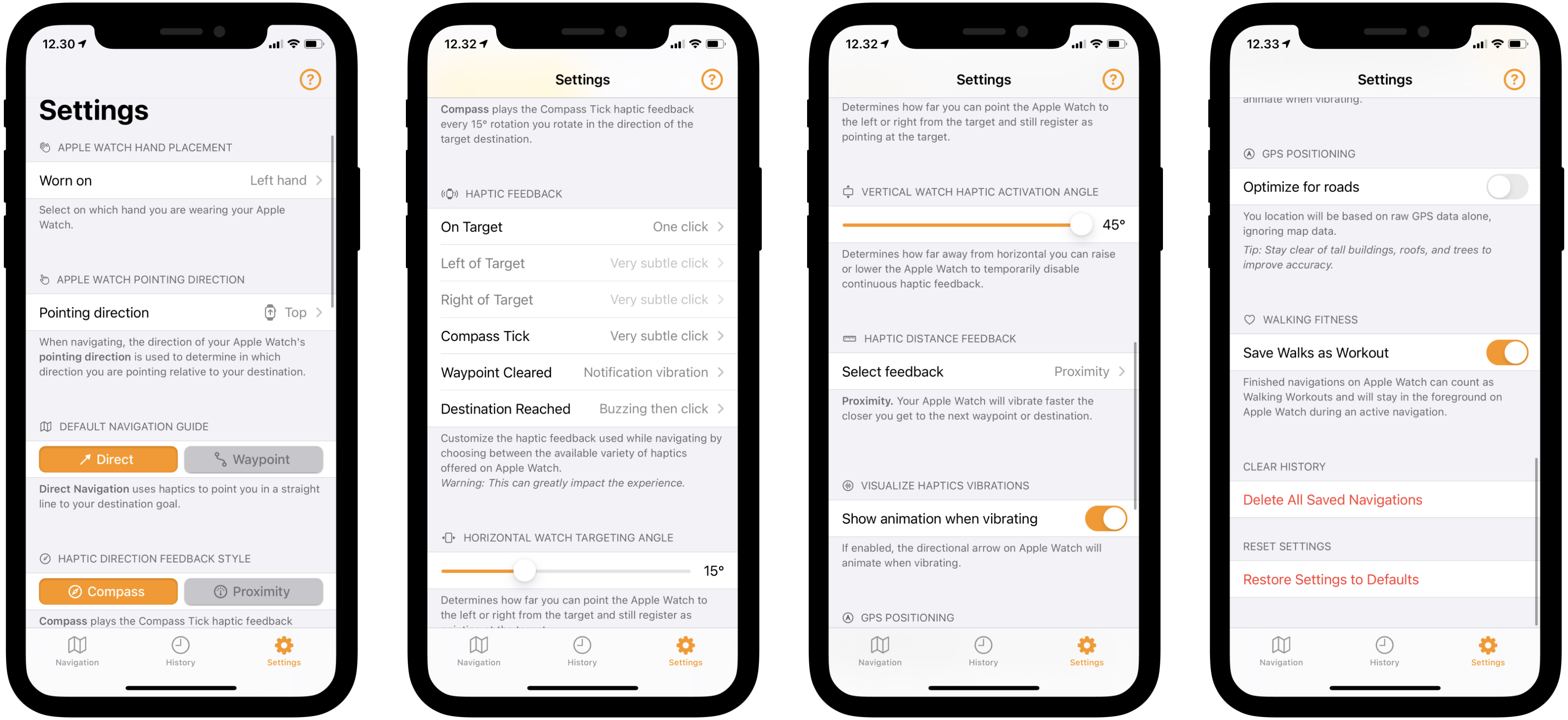


Figure 15: Scrolling the Settings screen and available options in Haptic Navigator.

Software Architecture

A Model–View–ViewModel (MVVM) software architecture was adopted throughout for compatibility with the SwiftUI framework to develop the front-end. To control the variety of data flows from different sensors and cross-platform wireless connectivity, the Combine framework was used extensively throughout for asynchronous event handling[3]. The project's ViewModel data structure is described in the figure below (see Figure 16), outlining the primary components that make up the functionality of the app to give some sense of the various parts that makes up the final prototype. These components consists of:

The **ConnectivityManager** classes manage the connectivity between the two platforms, including availability and reachability. The app will show if the other app is out of reach, not installed or if a watch is not even paired.

The **SyncManager** classes manage which data to send and which received data to keep or discard.

The **NavigationManager** classes manage the active navigation and its states, which can be either in setup, active, paused, or finished. On Apple Watch, this class is responsible for combining data from multiple sources (LocationManager, MotionManager, RoutePlanner, SettingsManager, etc.) and calculate the appropriate navigation state.

The **HapticManager** on Apple Watch observes the NavigationManager to calculate the appropriate haptic response.

The **WatchSearchProvider** and **WatchSearchRequester** handle search requests from the Apple Watch sent to the iPhone, which attempts to fetch search results and route data to return to the Apple Watch. This approach was used since route planning and querying the Apple map service are not available on WatchOS.

The **SettingsManager** maintains loading, updating, and saving changes to settings to persistent storage.

The **LocationManager** handles location services and data privacy controls related to user location. Location updates are only started and stopped when a navigation is active, but the user's location is also requested when searching for a destination, starting from the user's current position.

The **GeoCoder** is responsible for fetching human-readable address names and point-of-interest names from geographic coordinates.

The **MetricFormatters** is a group of classes responsible for converting various metrics to appropriate representations in the UI.

The **LocationSearchCompleter** monitors user typing and suggests location search results based on complete or partial search queries.

The **LocationSearcher** is responsible for fetching geographic coordinates based on location names or addresses.

The **RoutePlanner** is responsible for fetching route suggestions between two locations, including estimating distance and arrival time. It is also responsible for creating waypoints based on the route data.

3 Apple's Combine Framework https://developer.apple.com/documentation/combine

The **SearchManager** manages the LocationSearcher and Location-SearchCompleter as a coherent search experience, including the search regions to improve location searching to provide more relevant search results.

The **NavigationStore** is responsible for loading, sorting, deleting, and saving navigation records (recorded data during navigation) to persistent storage.

The **ComplicationController** handles Apple Watch complications. To integrate Haptic Navigator with the users existing watch face. Complications provide quick, glanceable information about the app's state when the app is not in the foreground. Complications are also used as a short-cut to open the app from the watch face.

The **HapticManager** manages haptic feedback, including timing and queuing of haptic patterns based on observed user preferences and navigation state (including motion and location state).

The **MotionManager** manages motion sensor readings, including gyroscope, accelerometer, and compass readings. Quaternion rotation is used to rotate two unit vectors corresponding to the watch's pointing direction and screen facing direction (perpendicular to the watch screen) and based on left or right-hand settings and watch pointing direction preferences. The MotionManager also manages user motion data privacy and is used to estimate pedometer data to calculate travel distance.

The **WorkoutManager** is used to manage Apple's HealthKit service, to record pedestrian "Workout" data in the users HealthStore if allowed. This is primarily used to gain performance and background execution benefits to the Apple Watch that would otherwise not be permitted for non-workout purposes (see Platform Limitations).

Platform Limitations

Although choosing a commercial platform for development has its benefits, several drawbacks using a closed system were also encountered. Although working under the constraints of the system used for development, learning these limitations and how to work with or around is an important aspect of the feasibility of a proposed solution. As an example, the Apple platform is strict on user data privacy. This has to be accounted for in the app, as handling users' location and motion data is privacy sensitive. The final prototype, therefore, requests permission to use this data. The data is only gathered if permitted by the user. The user is prompting once at startup with an option to accept or deny the use of this data, with a description of how and why the data is used. The app does not send any data to a third party, as all computations and navigation records are handled and stored locally on the user's device.

Haptic API Limitations

Using appropriate haptic patterns for the various interface features provided is critical to the overall experience of navigating using haptics. On the iOS platform, custom haptics can be created and used freely. However, on WatchOS (at the time of writing), using custom-designed haptic patterns is not possible, presenting a considerable constraint to the project. Instead, only a fixed set of twelve haptic patterns are available on Apple Watch, representing various feedback styles such as Success, Failure, Start, Stop, Click, etc. This selection of haptics is referred to programmatically as WKHapticType (Watch Kit Haptic Type), including information about each haptics intended purpose[4].

Since none of the available types provide a continuous vibration, they must be played together in short successions. However, it was not possible to find public information or documentation describing the technical details (such as duration) about each one. If a developer attempts to play these haptics in quick successions and too close to each other, the operating system imposes a minimum delay of 100 milliseconds before engaging the haptic engine again to generate the next feedback[5]. To work with these limitations and experiment with the haptic options available, a small utility app, WKHaptics (see Figure 17), was developed for this purpose alone.

Using WKHaptics, the following available types of patterns were examined when played in quick successions. The shortest rate at which each could be played reliable (without system intervention) was noted down and used in the final prototype (see Table 2).

*Since other developers or researchers might be faced with the same problem working with haptic vibrations on Apple Watch, the standalone app **WKHaptics** was made public as a free download[6].*

[4] Apple Developer Documentation: WKHapticType 🔗 https://developer.apple.com/documentation/watchkit/wkhaptictype

[5] Apple Developer Documentation: Play WKHaptic 🔗 https://developer.apple.com/documentation/watchkit/wkinterfacedevice/1628128-play

[6] App Store: WKHaptics 🔗 https://apps.apple.com/dk/app/wkhaptics/id1563451947

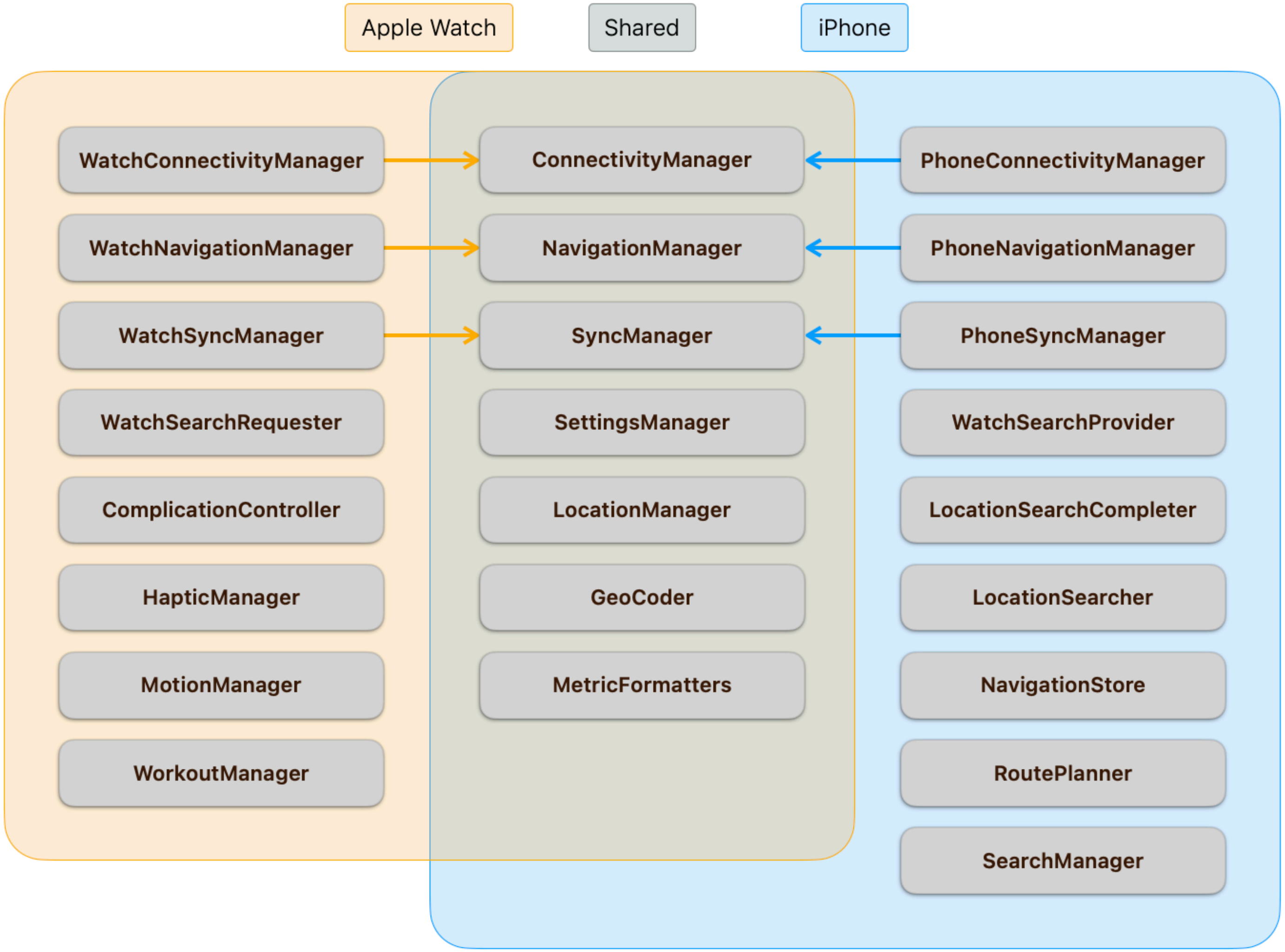


Figure 16: The *ViewModel* classes used in the final prototype.
The two boxes show classes used in the iPhone and Apple Watch apps respectively.
Their overlap shows classes that are shared. Class inheritance is visualized with arrows.

WKHapticType name	Description	Duration (in seconds)
Notification	Notification vibration	0.8
DirectionUp	Two quick clicks up	0.2
DirectionDown	Two quick clicks down	0.2
Success	Three quick clicks	0.4
Failure	Buzzing failure	0.4
Retry	Buzzing retry	0.4
Start	One click	0.2
Stop	Two clicks	0.7
Click	Very subtle click	0.1
NavigationLeftTurn	Three pairs of clicks	2.5
NavigationRightTurn	Twelve quick clicks	2.5
NavigationGenericManeuver	Buzzing then click	1.0

Table 2: The list of available types of haptic patterns on Apple Watch, including a subjective description and the minimum rate at which they can be played consecutively and reliably.

Another limitation was the lack of control of the strength (intensity) of haptics on the system. This is controlled by the user, in the Apple Watch Settings app, and can be set to either Default or Prominent. Users were suggested to change this to Prominent when testing the app on their own devices.

Figure 17: A screenshot from the WKHaptics app. The user can browse the available haptic patterns offered on Apple Watch and experience them played with different intervals that can be adjusted.

Findings

In this section, relevant findings are presented primarily related to results and observations from testing the final prototype, and conducting the research through design process including iterating through the design, development and testing process. Figures 18–21 show participants testing the prototype in various settings and contexts.

User Feedback and Observations

Learning Curve

After having introduced the participants to the concept of Haptic Navigator and getting them familiar with how to interpret the user interface, it was noted that although the basic idea of following the compass arrow to a destination seemed intuitive and simple to understand, learning to get comfortable with it was significantly more difficult for some of the participants. For two participants, although the visual confirmation of the arrow pointing direction was understood, the experience of the repeated haptic vibrations was first understood as a negative alert that should be avoided (instead of as a confirming interaction).

In general, the participants seemed to have high expectations about the system's accuracy, especially when using waypoint navigation. For example, if the arrow pointed into a wall too early before a turn or into an object that the participant could walk around, the participants would still be unsure about the system's intentions and if it was reliable. The participants could also follow the arrow too strictly and start jaywalking in the middle of the street if the arrow pointed slightly inwards from the side-

Figure 18: A participant in an urban setting in the moment of testing the right-pointing direction of the compass arrow.

Figure 19: A participant walking on a small path in the moment of testing the top pointing direction of the ↗ compass arrow.

walk towards the middle of the street. However, it was noted that with practice, the participants became better at making their own judgement about how to follow the arrow and get around obstacles.

As the participants got more experienced with the interface, checking the direction of the arrow using haptics alone also became significantly faster. Less experienced participants were observed to be more likely to stop in their tracks to scan and confirm their direction before walking again. As the participants became more experienced, they were more likely to quickly raise and lower the hand while walking to do a quick check of direction. Less experienced participants would check the interface more often, sometimes keeping their hand raised for longer while walking, compared to when they gained more experience and only checked briefly most of the time.

Eyes-Free Navigation

With practice, experienced participants were able to navigate using the haptic interface alone, although less confidently than when visuals were allowed. These attempts were made by the observer selecting locations secretly, such as using a seemingly random nearby street address that required multiple streets turns to reach. When reaching the destination, the participant would stop and announce that they had reached their final destination.

A considerable drawback of relying on the haptic interface for directions alone was that the participants had less of a sense of how far they were from finishing the navigation as they walking from waypoint to waypoint. The haptic distance indicator was only relevant up until the next waypoint or (depending on settings) only distinct and useful up to a shorter distance. Some users preferred switching it off entirely to reduce complexity in the interface. In contrast, others preferred it as an indicator of when reaching the next location (destination or waypoint) was imminent. This finding presents an argument for why directional guidance can be problematic when used in a vacuum and should preferably be supported by *distance guidance* for eyes-free haptic navigation.

When walking down long straight roads, the lack of a sense of remaining distance was also a concern, as participants could become increasingly uncertain if they were still on the right track or if they were misinterpreting the interface. One participant suggested getting an occasional vibration now and then to confirm the walking direction, also to not having to raise the wrist so often to reaffirm the walking direction. In contrast, another participant suggested introducing haptic feedback to alert if the participant was walking in the wrong direction. This was less of a problem when visuals were allowed as the participant could simply read the distance from the smartwatch screen, although requiring the user to look down while walking.

Social Acceptance

Using the pointing direction of the hand (as if pointing with the index finger to feel vibrations) was an easily understood mental model to use when learning the concept. But in practice, performing the required gesture to raise the wrist and point was considered impractical and socially awkward, as it required the participant to point in public or have their arm extended straight out from their body. One comment was that it resembled a Nazi salute. Instead, a more socially acceptable solution was to adjust the settings to point out from the top of the smartwatch screen.

Figure 20: A participant walking in an open field surrounded by trees following the directions using waypoint navigation.

Figure 21: A participant walking in an urban setting following the directions using the direct navigation mode.

This setup was the preferred choice for almost all participants, except one participant who preferred to point downwards in front of the body with a high enough vertical activation tolerance.

The issue of avoiding actions that are socially awkward was also observed when rotating the watch to scan for the target direction. It was noted that it was challenging to turn one's arm and upper body enough to scan for locations while walking, depending on the required amount of rotation. This meant that when the participants were checking for locations that were not directly in front of them, they would usually stop walking to more easily be able to rotate their bodies until they found the right direction to go. It was observed that waypoint navigation was more convenient in this regard, as reaching a waypoint (often at an intersection, turn, or crossing) gave a convenient opportunity to stop and scan for the next waypoint, which most of the time was a straight line walk.

Personal Preferences

The final prototype contains several options to modify the navigation experience and the behavior of the haptic interface — both for testing features and to adjust for personal preferences. As the participants experimented with various settings, it became clear that some can have very different opinions about what works best for them. The majority of participants preferred waypoint navigation, with one expressing the preference towards being guided as much as possible. Only one participant preferred direct navigation stating that it gave the participant more freedom to choose one's own way to the destination. Some participants also preferred the strongest possible haptics, while others were easily overwhelmed with it and preferred more subtle haptics.

A general preference towards a simple haptic interface with fewer types of haptic patterns to distinguish between was a reason that using the Compass over Proximity feedback style was preferred by most participants. One participant also chose to enable features like the distance indicator, while the majority preferred to keep the haptic interface as simple as possible to make the interface easier to interpret and quick to respond to.

In terms of differences in haptic sensitivity, it was noted that personal preferences with smartwatch sizes, wrist band materials, and how tight the smartwatch was around the wrist could play a significant role in this.

Technical Observations

Although most navigations were conducted without significant issues, technical difficulties were encountered mainly related to the accuracy of GPS and reliability problems with the magnetometer inside the Apple Watch.

A situation with a participant who wore a band containing magnets had a significant influence on the direction arrow in the prototype, making it unpredictable. For this reason, the participant had to use a different wristband for testing. The finding signifies the importance of being aware of the diversity and preferences of wearables and how they can influence each other.

Testing the prototype in an urban environment, walking against tall buildings and in narrow streets had a significant influence on GPS accuracy, often placing the user's estimated location incorrectly inside buildings they were walking past. Although walking in the middle of the street would theoretically improve the signal by moving away from the sidewalks and away from tall buildings, this was not practical. Experiments were done using the default location service for navigation available in Apple's Core Location framework, but this service attempted to estimate the user's location to road maps, occasionally moving the user's location down the wrong road. Instead, using raw GPS location readings results in the most reliable experience.

The issue with GPS accuracy problems in tight urban areas where less evident with direct navigation than when using waypoint navigation. This was primarily due to the longer distance to the final waypoint than to each waypoint along the destination route.

Conclusion

In this thesis, the design and development of a wrist-worn pedestrian navigation tool were presented, built around a novel approach of utilizing haptic technology, motion sensing, and location awareness for eyes-free navigation. The final prototype presents a design proposal for an end-user pedestrian navigation system running on a smartwatch and, at the same time, a research tool used to examine the developed design in realistic settings. Developers and researchers can use the tool to perform their own exploration of how haptics can be used for navigational purposes.

Following a research through design methodology, the project describes critical design and development considerations, followed by feedback gathered from participants that tested the prototype during various stages of the project. The end product is a body of knowledge contained in a piece of software that suggests further examinations, including some of the tools to perform these examinations. It is the hope of the author that the project can inspire other developers and designers to think outside the limitations of screen-based interactions and explore alternative futures where the digital can become both accessible and invisible.

Bibliography

Barnard, L., Yi, J. S., Jacko, J. A., & Sears, A. (2006). Capturing the effects of context on human performance in mobile computing systems. *Personal and Ubiquitous Computing*, *11*(2), 81-96. https://doi.org/10.1007/s00779-006-0063-x

Biswas, S., & Visell, Y. (2021). Haptic Perception, Mechanics, and Material Technologies for Virtual Reality. *Advanced Functional Materials*. https://doi.org/10.1002/adfm.202008186

Brodersen, G. (2019). A Sense of Direction: How the Use of Haptic Feedback in a Wrist-Worn Wearables Can Be Utilized for Spatial Directional Stimuli. https://brodersen.github.io/BachelorThesis/GabrielBrodersen-BachelorThesis2019.pdf

Cammann, H., Stolz, P., & Wölfel, M. (2017). Non-Visual Pedestrian Navigation using Vibrotactile Stimuli in the Wristband of a Smartwatch.

Cholewiak, R. W., & Collins, A. A. (2003). Vibrotactile localization on the arm: effects of place, space, and age. *Percept Psychophys*, *65*(7), 1058-1077. https://doi.org/10.3758/bf03194834

Craig, J. C. (1993). Anomalous sensations following prolonged tactile stimulation. *Neuropsychologia*, *31*(3), 277-291. https://doi.org/https://doi.org/10.1016/0028-3932(93)90092-E

Diggelen, F., & Enge, P. (2015). The World's first GPS MOOC and Worldwide Laboratory using Smartphones.

Dobbelstein, D., Henzler, P., & Rukzio, E. (2016). *Unconstrained Pedestrian Navigation based on Vibro-tactile Feedback around the Wristband of a Smartwatch* Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, San Jose, California, USA. https://doi.org/10.1145/2851581.2892292

Downs, R. M., & Stea, D. (1973). Image and environment: cognitive mapping and spatial behavior. Aldine Pub. Co.

Fröhlich, P., Oulasvirta, A., Baldauf, M., & Nurminen, A. (2011). On the move, wirelessly connected to the world. *Communications of the ACM*, *54*(1), 132-138. https://doi.org/10.1145/1866739.1866766

Gartner, G., Huang, H., Millonig, A., Schmidt, M., & Ortag, F. (2011). Human-centred Mobile Pedestrian Navigation Systems. *Mitteilungen der Österreichischen Geographischen Gesellschaft*, *153*, 237-250. https://doi.org/10.1553/moegg153s237

Ghouaiel, N., Cieutat, J.-M., & Jessel, J. P. (2013, 2013-02-24). Haptic System for Eyes Free and Hands Free Pedestrian Navigation. [Proceedings of the ixth International Conference on Advances in Computer-Human Interactions]. ACHI 2013 : The Sixth International Conference on Advances in Computer-Human Interactions, France.

Guo, W., Ni, W., Chen, I., Ding, Z. Q., & Yeo, S. H. (2009, 19-23 Dec. 2009). Intuitive vibro-tactile feedback for human body movement guidance. 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO),

Heikkinen, J., Olsson, T., & Väänänen, K. (2009). Expectations for User Experience in Haptic Communication with Mobile Devices. https://doi.org/10.1145/1613858.1613895

Heuten, W., Henze, N., Boll, S., & Pielot, M. (2008). Tactile wayfinder: A non-visual support system for wayfinding (Vol. 358). https://doi.org/10.1145/1463160.1463179

Hong, J., Pradhan, A., Froehlich, J. E., & Findlater, L. (2017). *Evaluating Wrist-Based Haptic Feedback for Non-Visual Target Finding and Path Tracing on a 2D Surface* Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, Baltimore, Maryland, USA. https://doi.org/10.1145/3132525.3132538

Hong, J., Stearns, L., Froehlich, J., Ross, D., & Findlater, L. (2016). *Evaluating Angular Accuracy of Wrist-based Haptic Directional Guidance for Hand Movement* Proceedings of the 42nd Graphics Interface Conference, Victoria, British Columbia, Canada.

Hyman Jr, I. E., Boss, S. M., Wise, B. M., McKenzie, K. E., & Caggiano, J. M. (2010). Did you see the unicycling clown? Inattentional blindness while walking and talking on a cell phone. *Applied Cognitive Psychology*, *24*(5), 597-607. https://doi.org/https://doi.org/10.1002/acp.1638

Jacob, R., Mooney, P., Corcoran, P., & Winstanley, A. (2011). Integrating haptic feedback to pedestrian navigation applications.

Jacob, R., Winstanley, A., Togher, N., & Roche, R. (2012). Pedestrian navigation using haptic feedback: Results from a field study to test spatial abilities and memory recall.

Jones, L. A. (2017). A Haptic Compass for Navigation. *Computer*, *50*(6), 6-6. https://doi.org/10.1109/mc.2017.178

Kerdegari, H., Kim, Y., & Prescott, T. J. (2016). Head-Mounted Sensory Augmentation Device: Designing a Tactile Language. *IEEE Trans Haptics*, *9*(3), 376-386. https://doi.org/10.1109/toh.2016.2554111

Keyson, D., & Bruns Alonso, M. (2009). Empirical Research Through Design.

Laurier, E., & Brown, B. (2008). Rotating Maps and Readers: Praxiological Aspects of Alignment and Orientation. *Transactions of the Institute of British Geographers*, *33*, 201-216. https://doi.org/10.1111/j.1475-5661.2008.00300.x

Lichenstein, R., Smith, D. C., Ambrose, J. L., & Moody, L. A. (2012). Headphone use and pedestrian injury and death in the United States: 2004-2011. *Inj Prev*, *18*(5), 287-290. https://doi.org/10.1136/injuryprev-2011-040161

Maclean, K., & Enriquez, M. (2003). Perceptual design of haptic icons. *In Proceedings of EuroHaptics*.

Ng, J. Y. C., & Man, J. C. F. (2004). Vibro-Monitor: A Vibrotactile display for Physiological Data Monitoring.

O'Modhrain, S. (2004). Touch and Go — Designing Haptic Feedback for a Hand-Held Mobile Device. *BT Technology Journal*, *22*(4), 139-145. https://doi.org/10.1023/B:BTTJ.0000047592.21315.ce

Oakley, I., Kim, Y., Lee, J., & Ryu, J. (2006). Determining the Feasibility of Forearm Mounted Vibrotactile Displays (Vol. 2006). https://doi.org/10.1109/VR.2006.49

Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 919–928). Association for Computing Machinery. https://doi.org/10.1145/1054972.1055101

Oviatt, S. (1999). Ten myths of multimodal interaction. *Communications of the ACM*, *42*(11), 74-81. https://doi.org/10.1145/319382.319398

Panëels, S., Anastassova, M., Strachan, S., Van, S. P., Sivacoumarane, S., & Bolzmacher, C. (2013, 14-17 April 2013). What's around me? Multi-actuator haptic feedback on the wrist. 2013 World Haptics Conference (WHC),

Panëels, S., Brunet, L., & Strachan, S. (2013). Strike a Pose: Directional Cueing on the Wrist and the Effect of Orientation (Vol. 7989). https://doi.org/10.1007/978-3-642-41068-0_13

Pielot, M., Henze, N., Heuten, W., & Boll, S. (2008). Evaluation of Continuous Direction Encoding with Tactile Belts (Vol. 5270). https://doi.org/10.1007/978-3-540-87883-4_1

Pielot, M., Poppinga, B., & Boll, S. (2010a). PocketNavigator: using a Tactile Compass to Enhance Everyday Pedestrian Navigation Systems. *Proceedings of MobileHCI, Lisboa, Portugal*.

Pielot, M., Poppinga, B., & Boll, S. (2010b). PocketNavigator: vibro-tactile waypoint navigation for everyday mobile devices. https://doi.org/10.1145/1851600.1851696

Pielot, M., Poppinga, B., Heuten, W., & Boll, S. (2011a). 6th senses for everyone! The value of multimodal feedback in handheld navigation aids. https://doi.org/10.1145/2070481.2070496

Pielot, M., Poppinga, B., Heuten, W., & Boll, S. (2011b). *A Tactile Compass for Eyes-Free Pedestrian Navigation* (Vol. 6947). https://doi.org/10.1007/978-3-642-23771-3_47

Pielot, M., Poppinga, B., Heuten, W., & Boll, S. (2012). PocketNavigator: Studying Tactile Navigation Systems In-Situ. *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA*. https://doi.org/10.1145/2207676.2208728

Rawassizadeh, R., Price, B. A., & Petre, M. (2014). Wearables: has the age of smartwatches finally arrived? *Commun. ACM*, *58*(1), 45–47. https://doi.org/10.1145/2629633

Renaudin, V., Dommes, A., & Guilbot, M. (2017). Engineering, Human, and Legal Challenges of Navigation Systems for Personal Mobility. *IEEE Transactions on Intelligent Transportation Systems*, *18*(1), 177-191. https://doi.org/10.1109/tits.2016.2563481

Rimol, M. (2021). *Gartner Forecasts Global Spending on Wearable Devices to Total $81.5 Billion in 2021*. Gartner. Retrieved August 23. from https://www.gartner.com/en/newsroom/press-releases/2021-01-11-gartner-forecasts-global-spending-on-wearable-devices-to-total-81-5-billion-in-2021

Robinson, S., Jones, M., Eslambolchilar, P., Murray-Smith, R., & Lindborg, M. (2010). *"I Did It My Way": Moving Away from the Tyranny of Turn-by-Turn Pedestrian Navigation*. https://doi.org/10.1145/1851600.1851660

Rodríguez, J.-L., Velázquez, R., Del-Valle-Soto, C., Gutiérrez, S., Varona, J., & Enríquez-Zarate, J. (2019). Active and Passive Haptic Perception of Shape: Passive Haptics Can Support Navigation. *Electronics*, *8*(3). https://doi.org/10.3390/electronics8030355

Schätzle, S., Hulin, T., Preusche, C., & Hirzinger, G. (2006). *Evaluation of Vibro-Tactile Feedback to the Human Arm*.

Schön, D. (1983). The reflective practitioner: How professionals think in action. Ashgate Publishing.

Sergi, F., Accoto, D., Campolo, D., & Guglielmelli, E. (2008). Forearm orientation guidance with a vibrotactile feedback bracelet: On the directionality of tactile motor communication. https://doi.org/10.1109/BIOROB.2008.4762827

Sreelakshmi, M., & Subash, T. D. (2017). Haptic Technology: A comprehensive review on its applications and future prospects. *Materials Today: Proceedings*, *4*, 4182-4187. https://doi.org/10.1016/j.matpr.2017.02.120

Tsukada, K., & Yasumura, M. (2004). ActiveBelt: Belt-Type Wearable Tactile Display for Directional Navigation (Vol. 44). https://doi.org/10.1007/978-3-540-30119-6_23

Van Erp, J. (2005). Presenting directions with a vibrotactile torso display. *Ergonomics*, *48*(3), 302-313. https://doi.org/10.1080/0014013042000327670

van Erp, J. B. F. (2001, 2001//). Tactile navigation display. Haptic Human-Computer Interaction, Berlin, Heidelberg.

Weber, B., Schätzle, S., Hulin, T., Preusche, C., & Deml, B. (2011, 21-24 June 2011). Evaluation of a vibrotactile feedback device for spatial guidance. 2011 IEEE World Haptics Conference, Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. https://doi.org/10.1145/1240624.1240704