GesturePod: Enabling On-device Gesture-based Interaction for White Cane Users

Shishir G. Patil Microsoft Research India
Don Kurian Dennis Microsoft Research India
Chirag Pabbaraju Microsoft Research India
Nadeem Shaheer Microsoft Research India
Harsha Vardhan Simhadri Microsoft Research India
Vivek Seshadri Microsoft Research India
Manik Varma Microsoft Research India
Prateek Jain Microsoft Research India

ABSTRACT

People using white canes for navigation find it challenging to concurrently access devices such as smartphones. Building on prior research on abandonment of specialized devices, we explore a new touch free mode of interaction wherein a person with visual impairment can perform gestures on their existing white cane to trigger tasks on their smartphone. We present GesturePod, an easy-to-integrate device that clips on to any white cane, and detects gestures performed with the cane. With GesturePod, a user can perform common tasks on their smartphone without touch or even removing the phone from their pocket or bag. We discuss the challenges in building the device and our design choices. We propose a novel, efficient machine learning pipeline to train and deploy the gesture recognition model. Our in-lab study shows that GesturePod achieves 92% gesture recognition accuracy and can help perform common smartphone tasks faster. Our in-wild study suggests that GesturePod is a promising tool to improve smartphone access for people with VI, especially in constrained outdoor scenarios.

Author Keywords

Gesture recognition; Visual impairment; Smartphone access; White cane; Resource constrained machine learning

CCS Concepts

•**Human-centered computing** → **Interaction devices;** *Ubiquitous and mobile computing; Ubiquitous and mobile computing systems and tools; Accessibility systems and tools;*

INTRODUCTION

Smartphones have become an integral part of our lives. While new technologies and applications on the smartphone have improved the lives of all people, smartphones have significant potential to positively impact the lives of people with visual impairments (VI). Mainstream apps such as ride-hailing and maps are beginning to combat the accessibility barriers. Furthermore, recent smartphone apps such as Seeing.AI [33], Soundscape [32], and Eye-D [42] allow people with VI to be more aware of and better navigate their surroundings. These apps provide a glimpse of the exciting possibilities that smartphones can offer people with VI.

Despite these benefits, smartphone accessibility remains a challenge. A prior study [22] has shown that the user interfaces for mobile apps can be cumbersome to use for people with VI. Furthermore, for users with VI who use a cane for navigation, there are many situations where even accessing or locating the smartphone in a timely manner can be difficult. Figure 2a shows one such situation where both the hands of a person with VI are occupied - one hand with a cane and the other hand with a coffee mug. In this paper, we focus on this problem of accessibility to a smartphone in constrained settings. We particularly emphasize on the Global South, where people living with disabilities often have lower incomes [36]

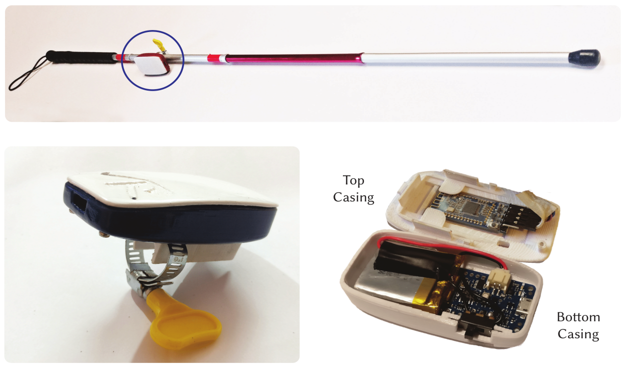


Figure 1: GesturePod. Top image shows the pod attached to a white cane. Bottom left image shows the close-up view of the pod. Bottom right image shows the interior of the pod.

GesturePod code available at https://aka.ms/GesturePod
For readers interested in learning more details of the ML technology underlying our work, refer to our technical report [6].

and typically cannot afford expensive accessibility technologies, e.g., smart watch.

One way to improve access to the phone without any additional devices is to use voice commands. Unfortunately, voice commands are not always effective (e.g., noisy environments) and not always desirable due to privacy reasons [48]. In addition, many languages in the world do not yet have good voice command support. On the other hand, prior works have proposed wearable devices such as rings [1] and dials [23] to improve access to the phone. However, such devices are not effective when both hands of the user are occupied.

Given that most people with VI use a white cane for navigation, prior works have proposed to augment the white cane with buttons [4] or a touchpad [44] to allow a person with VI to more easily interact with their smartphone. However, these solutions significantly modify the cane handle and require the user to change their normal grip, which is highly not desirable. Moreover, touchpad-based solutions significantly increase the weight and cost of the the white cane. In an informal conversation, Saqib Shaikh, the founder of Seeing.AI [33] and a white cane user himself, commented:

I am not a fan of devices which put the electronics in the cane handle, making it heavy and bulky. I like the ergonomics of my existing cane. ... Since I like to change my cane every year or so, I would prefer a device that can be easily moved to a new cane.

In this work, we explore a new mode mode of converting the white cane into an interaction device that is complementary to prior approaches—*performing gestures using the cane*. As gestures are natural, and relatively easy to learn and perform, they offer a promising avenue for interaction. However, for such a solution to be practical, we must 1) design the gestures such that they do not interfere with the normal can usage, 2) ensure that the solution can be adapted to any cane, 3) ensure minimal increase in the weight of the cane, 4) any device lasts at least an entire day, and 5) the gestures are recognized in a wide variety of settings (e.g., users, flooring). To cater to users in low-income settings, we also require the cost of the solution to be as low as possible.

The main contribution of this work is the GesturePod, a device that can be clamped on to *any* white cane making the cane a gesture-based interaction device (Figure 1). To reduce cost and weight, GesturePod uses a low-cost, lightweight microcontroller, off-the-shelf sensors (accelerometer + gyroscope), a BLE module for communication, and a small battery. We design five gestures that are intuitive and do not interfere with normal cane usage (described in Figure 2b-f). To reduce power consumption and robustly recognize the gestures, we carefully designed a machine learning (ML) pipeline that allows GesturePod to run the gesture recognition *fully* on the microcontroller. Overall, GesturePod costs less than 10 USD, weighs 49 g, continuously runs for 28 hours on a single charge and accurately recognizes the gestures in the real world across a range of users and environments.

To enable deployment of our ML algorithm on the microcontroller, we designed novel features and exploit recently proposed ProtoNN algorithm [13]. Our ML pipeline consists of 1) a simple methodology to collect training data for gestures, 2) tools to automatically curate the collected data and train the ML model, and 3) an optimized code that runs the trained model on the microcontroller. This pipeline allows addition of new gestures or replacement of existing gestures. For example, two high-school students added a sixth gesture to our GesturePod after playing with it for a day.

Based on exploratory interviews, we designed an Android app that maps gestures on the cane to common smartphone tasks. To understand the usefulness and adaptability of GesturePod based cane, we conducted two sets of user studies: an in-lab study, and an in-wild study. Our user studies indicate that 1) users learn to use the GesturePod with just 10 minutes of training, 2) it can detect gestures with 92% accuracy across multiple users and environments, and 3) it can significantly improve the time to perform common smartphone tasks, especially in constrained settings.

Our paper makes the following contributions.

- We design GesturePod, an easy-to-integrate device that can be clamped on to any white cane and recognize a range of gestures performed on the cane. We solve several technical challenges to make GesturePod robust, power-efficient, lightweight, and inexpensive.
- We develop an ML pipeline to detect easy-to-perform yet non-trivial-to-recognize gestures in many environments on battery-powered microcontroller.
- Our in-lab user study indicates that GesturePod can accurately recognize gestures across a range of users and settings, and significantly reduce the time to complete common smartphone tasks.
- Feedback from users in our in-wild study indicates that using gestures on the white cane is a promising method for interacting with smartphone.

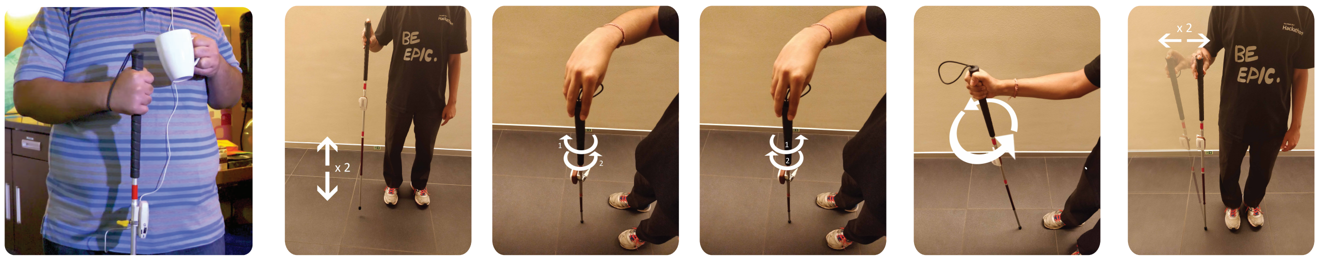
RELATED WORK

The goal of this work is to design a lightweight, low-cost device that can recognize gestures performed using a white cane. We divide related work into two categories: those that enhance the white cane with different goals, and those that propose gesture recognition.

Enhancements to the White Cane

Prior works have observed that the white cane is an important navigation device for people with VI [51, 45]. Williams et al. [46] extensively study various navigational challenges for people with VI. Kim et al. [18] highlight several key issues such as battery life and reaction time that one must consider when augmenting the white cane with electronics.

There are two closely related works to ours. The first is that by Batterman et al. [4], who propose to augment the cane with a set of buttons to allow users to trigger actions on their smartphone. The second work is WeWalk [44] that replaces the cane handle with one that contains a touchpad. Both these solutions require the user to significantly change their grip



(a) Constrained environment (b) Double Tap (c) Right Twist (d) Left Twist (e) Twirl (f) Double Swipe

Figure 2: (a) A visually impaired person in a constrained environment (both hands occupied). (b)-(f) Illustration of various gestures used in our study.

before performing the actions required to access their smartphone, which is not desirable. In fact, our informal discussions with potential users indicate that the they prefer touch-free interaction mechanisms like gestures.

Other prior works have added sensors like ultrasound sensors [39], camera [17], LIDAR [37], RFID tags [7, 11, 25], etc. to assist users with VI to detect obstacles. Connected Cane [40] augments the cane with sensors to enable fall detection and alert people in case of a fall. Some studies [10, 3, 20] augment the cane with IMU sensors (accelerometer + gyroscope) mainly to understand the usage pattern of the cane. These works do not aim to use the cane as an interaction device to access the smartphone.

Our approach of using gestures on the cane to interact with the smartphone addresses many of the challenges with prior works. Evaluation results from our studies validate this claim. Having said that, our solution can be easily integrated with almost all these prior approaches to provide a richer experience for people with VI.

Gesture Recognition

Gesture recognition is an extensively-studied problem, especially in the context of touch-free interaction systems [2, 24, 47]. However, most existing gesture recognition solutions for accessibility use vision-based techniques [35, 38, 15, 34, 12]. In addition to incurring high cost and power consumption, these solutions require addition of cameras at specific orientations, thereby making them impractical in our scenario.

Prior works have proposed mechanisms to use gestures on hand-based wearables as an interaction mechanism [8, 14, 21, 26, 31, 19, 29]. These solutions use IMU sensors in wearables such as a watch [28], electromyography and accelerometer sensors on the hand [27], or a finger ring with IMU sensors [16]. Unfortunately, these solutions are ineffective for users with VI particularly in settings where both their hands are occupied.

One key aspect of gesture recognition is the algorithm used for classifying gestures. Prior works have used hand-tuned rules to detect activities like running, sleeping [9], simple ML algorithms to recognize simple gestures in restricted settings [27], or expensive ML algorithms that are run on a powerful device [26]. However, none of these approaches suit our setting. On the one hand, rule-based algorithms or simple ML models cannot robustly detect our set of gestures that are simple to perform yet complex to recognize. On the other hand, more powerful ML models [49, 30] will require our device to transmit all the signals to a more powerful device (e.g., the smartphone). As we will describe shortly, this approach drains the battery on the device within a couple of hours, making it impractical.

DESIGN PRINCIPLES AND CHALLENGES

In this section, we first describe the rationale behind our approach of using gestures on the cane to interact with the phone. We then describe the key design constraints and challenges in making this approach practical in the real world.

Why Gestures on the Cane?

Why Gestures?

There are several reasons why gestures are better than other modes of interaction such as buttons or touchpads. First, as gestures are a natural mode of interaction, *they are easy to learn and remember*. In fact, all the users in our studies picked up the gestures with just ten minutes of training. Second, *gestures are easy to perform*. Mechanisms like buttons or touchpad require users to move their hands along the cane to first locate the device and then perform the necessary action. In contrast, users can perform gestures on the cane without even having to significantly change the grip with which they are holding the cane. Third, unlike buttons, adding new gestures only requires changes to the software and not the hardware.

Why on the Cane?

Besides the cane, locations that possibly have sufficient degrees of freedom to perform a range of gestures are the user's hands, fingers, head, and legs. As mentioned before, performing gestures using hands and fingers is not feasible especially when both the user's hands are holding objects. Performing gestures using the head or legs can potentially result in strain and sometimes even be dangerous. As prior works have observed [45, 50], the cane is a widely-used navigation device by users with VI. As a result, using the cane to perform gestures was a natural choice.

Design Constraints and Implications

Our goal is to design a gesture-recognition device that can be clamped on to any white cane without requiring further modifications to the cane. There are four constraints that make the design of such a device technically challenging.

1. Low Weight. As the device will be mounted on the cane, a large increase in the weight of the cane will affect the user's regular cane usage. As a result, the device must be of low weight. Consequently, we cannot simply mount any heavy-weight device like a smartphone to the cane. This constraint also limits the size of the battery.

2. Day-long Operation. Similar to other battery-operated electronic devices, we can only expect the user to put our device to charge at the end of the day. With limited battery capacity, this constraint essentially translates to low power consumption. This not only limits us to low-power consuming devices, it also eliminates the option of transferring all the sensor data to the smartphone. The communication alone drains the battery within a couple of hours.

3. Low Cost. As the focus of this work is on designing a solution that works for low-income users, the cost of the device has to be low. This constraint restricts us to low-cost microcontrollers and off-the-shelf sensors.

4. Robust Gesture Recognition. Finally, for the device to be practical, it must robustly recognize the set of gestures under a range of environments and users. This means *low false positives* as they can trigger unnecessary actions on the smartphone and *low false negatives* as high false negatives can make the system unusable. This constraint eliminates the option of using simple rule-based systems. In fact, even after a month of effort, we were unable to create a rule-based system with low false positives and low false negatives.

Technical Challenge

The main conclusions from the above constraints are as follows. First, our device can only use simple microcontrollers that incur low cost and power consumption. Second, since the sensor data cannot be transmitted to the smartphone due to battery limitations, the only option is to run the gesture recognition algorithm on the microcontroller itself and transmit only recognized gestures to the smartphone. Third, microcontroller have very small amounts of compute and memory resources. Therefore, as an added implication of the previous point, we cannot use standard ML classifiers as they require large amount of compute and memory resources to make real-time predictions.

So, the key technical challenge that we address is:

> Can we design a low-cost, low-power device that accurately and robustly recognizes a number of natural gestures in real-time using just a low-end microcontroller and off-the-shelf sensors?

In the following section, we describe the design and implementation of GesturePod that successfully addresses the above challenge.

GESTUREPOD: DESIGN AND IMPLEMENTATION

GesturePod is a plug-and-play device that can be clipped onto any white cane and pairs with Android phones. The microcontroller on board runs a machine learning model that infers

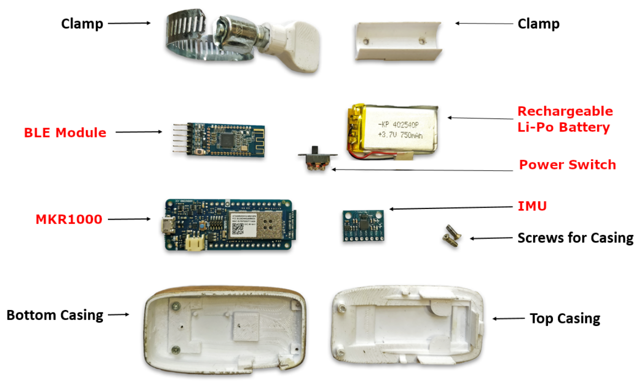


Figure 3: Components of GesturePod.

cane gestures performed by the user in real-time. The gestures are communicated to the phone where they invoke specific tasks (e.g. read out the time). The pod is also used for collecting labelled data necessary for training the ML model. We now describe design of the hardware and software components of GesturePod.

Components of GesturePod

GesturePod consists of five key electronic components (highlighted in red in Figure 3): 1) an off-the-shelf Inertial Measurement Unit (IMU) — MPU6050, 2) an Arduino MKR1000 board, 3) a Bluetooth Low-Energy (BLE) module, 4) a rechargeable battery, and 5) an on-off switch. The IMU contains an accelerometer and a gyroscope that capture data at a frequency of 200 Hz, and an internal buffer that can store 100 ms worth of sensor data. The MKR1000 board consists of an ARM Cortex-M0+ microcontroller with 32 KB working memory and 256 KB of read-only flash. The microcontroller runs our entire gesture recognition pipeline. GesturePod uses the BLE module to communicate recognized gestures to a connected smartphone. The rechargeable battery has 750 mAh capacity and powers all the above components. All these components are housed in a 76 mm×38 mm×25 mm casing with a clamp that allows us to mount GesturePod on any white cane.

Proposed Gestures on the Cane

We designed our gestures such that a) they are easy to perform, b) they are not accidentally triggered during natural cane use, and c) they map to common navigation tools for accessibility typically found in Android devices. Specifically, we use the *double swipe* gesture for the *select* action, *right twist* for *next*, *left twist* for *previous*, and *double tap* for *back/exit*. For added functionality, we included a fifth gesture: *twirl*. We did not include single tap as a gesture as it gets triggered inadvertently too often during normal usage. The following list describes how each of these gestures are performed (pictorially shown in Figure 2).

1. **Double tap** (D-T): Tap the cane on the floor twice
2. **Right twist** (R-T): Twist the cane to the right
3. **Left twist** (L-T): Twist the cane to the left
4. **Double swipe** (D-S): Tilt the cane to the right twice
5. **Twirl** (Tw): Make a circle with the cane's grip

While in this work, we focus on the above mentioned 5 gestures, adding a new gesture and/or replacing an existing gesture only requires a) collecting data points for the new gesture, and b) training a new model using our ML pipeline. We describe these steps in more detail in the following sections.

Data Collection for Model Training

To train our ML model, we need: 1) positive examples for each of the five gestures, and 2) negative examples where no gesture is performed. We used two different methods to collect, curate, and augment training data for each kind.

Positive Examples for Gestures

Seven sighted volunteers helped us collect training data for the five gestures. For each gesture, the volunteer performs the gesture using the cane mounted with the GesturePod, while an observer roughly marks the boundaries of the gesture in a program running on the computer. The program collects the sensor data using serial communication, labels it with the corresponding gesture, and stores it in a database. To ensure robust gesture recognition, our training data included variations in flooring, grip, orientation of the cane, and handedness. In total, we collected data for 102 double taps, 55 right twists, 54 left twists, 353 twirls, and 83 double swipes.

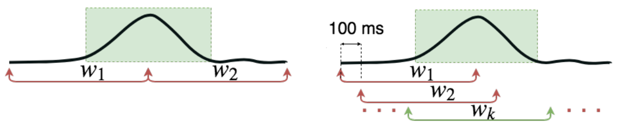
As the observer only approximately marked the boundaries of each gesture, we further curated the training data to ensure same duration for each example. We observed that all the gestures could be performed within 1.5 seconds. So, we manually trimmed our training examples such that 1) each example had sensor data for exactly 2 seconds, and 2) the gesture is roughly centered within the 2-second window. We increase the training set size by ten-folds by adding additional examples where the region containing the gesture is shifted on either side by up to 25 ms at 5 ms steps.

Negative Examples

For the GesturePod to be usable, we must ensure that it does not falsely trigger any of our gestures during the normal use of the cane. While one can easily design a rule-based system to distinguish between our five gestures, engineering a rule-based system to avoid false positives in the numerous scenarios occurring in regular cane use is impractical. In fact, to significantly reduce the false positive rate in our ML-based model, we had to add three kinds of negative examples to our training dataset.

First, we added negative examples from regular cane use without any gestures. For this purpose, we connected GesturePod to an SD-card to record *all* sensor data from the IMU. We attached the pod to a cane and asked our volunteers to walk around with the cane without performing any of the five gestures. We collected 8 minutes of such data and clipped it to generate 2-second segments of negative examples.

Our initial model trained with just these negative examples resulted in high false positive rate. One of the major source of false positives was partial gestures, e.g., a single tap, that were recognized by the GesturePod as a full gesture. We found that these partial gestures occur frequently during natural cane use. For instance, the "three-point" technique [41] to climb stairs involves a single tap on each step. To avoid these false positives, we retrained our model using negative examples augmented with data from partial gestures, e.g., single tap, half a twist, etc.



(a) Disjoint windows **(b) Overlapping windows**

Figure 4: Striding Windows— w_1 and w_2 are disjoint windows in (a), while they are overlapping in (b). The 2s window that contains the gesture is highlighted.

For the last round of data collection, we tagged any additional false positives generated by our model. For each instance, we extracted the corresponding sensor data and included it in our dataset as a negative example.

Machine Learning Pipeline

Our goal is to train a ML model that can take a continuous stream of data from the IMU sensors and detect occurrences of gestures. However, similar to the training examples, we cannot simply segment the sensor data stream into disjoint 2-second windows and run the model on each window. This is because a gesture performed by the user may cut across two consecutive windows as shown in Figure 4a. To mitigate this issue, we generate overlapping 2-second windows by sliding windows by 100 ms so that there is at least one window that encompasses the full gesture—e.g., in Figure 4b, window W_k contains the full gesture.

At the beginning of every sliding window (i.e. every 100 ms), the microcontroller must 1) fetch the IMU sensor data for the past 100 ms, 2) run the ML prediction on the latest 2-second window, and 3) communicate any recognized gesture to the smartphone. Steps (1) and (3) consume roughly 20 ms. Therefore, ML prediction must complete within 80 ms.

An ML predictor typically consists of two parts: 1) data featurization that converts raw sensor data into features that are suitable for the ML model, and 2) a classification algorithm that classifies the featurized data into one of the gestures (including no gesture).

For our classification algorithm, we use the multi-class formulation of the recently-proposed ProtoNN [13] algorithm which is specifically designed to generate models small enough to run on microcontrollers. However, when we employed ProtoNN with standard ML features (e.g., FFT features [5], clustering-based features [43]), it exceeded the time budget due to data featurization cost. Therefore, we had to design a set of features that 1) are easy enough to compute, 2) consumes small amount of memory, and 3) are still robust enough to be able to discern practical gestures.

Our Proposed Features

As mentioned before, the featurization step converts raw sensor data into a set of features. The raw sensor data consists of six dimensions, three each from the accelerometer and the gyroscope. For 2-seconds, when sampled at 200 Hz, the raw

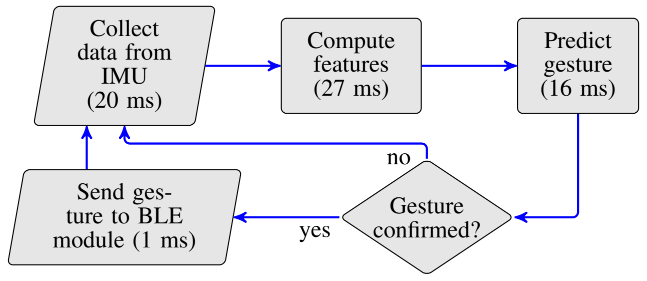


Figure 5: Prediction pipeline loop that runs on the single-threaded microcontroller (must finish in 100 ms).

data for each prediction instance consist of 400 values in each of the six dimensions.

We design two kinds of features. First, for each of the 6 dimensions, we group the 400 values into 20 equally-spaced bins in their range and count the number of values in each bin. Such equally-spaced bin counts are particularly efficient to compute in our striding-window setting. Second, bin count features discard phase in the gyroscope values (clockwise vs. anti-clockwise). Specifically, we need the phase information along the vertical axis of the cane to distinguish between the two twist gestures. To capture this information, we add four additional features: the index and length of the longest positive and negative sequence of gyroscope values along the vertical axis of the cane. In total, we compute 124 features for each training sample.

Model Training

With these new set of features, we trained a ProtoNN model on our dataset using a commodity PC. We randomly split the collected training dataset into 80% training samples and 20% testing samples. We tuned the ProtoNN hyper-parameters simultaneously to achieve high accuracy and low model size. Our final model is just 6 KB in size and achieves an accuracy of 99.9% on the test data.

Prediction Pipeline

The gesture prediction pipeline runs a continuous cycle of (a) data collection from IMU, (b) feature computation, (c) ProtoNN inference algorithm on the model generated by the ProtoNN training algorithm and (d) BLE communication for relaying the gestures detected to the phone. With our new feature set and the 6 KB ProtoNN model, the microcontroller can complete data featurization and the ML classification in 27 ms and 16 ms, respectively.

To reduce false positives, we use a secondary filter wherein our algorithm keeps track of the latest n predictions. If the majority of the these predictions point to a particular gesture, then our algorithm *confirms* the presence of that gesture and communicates the gesture to the smartphone. Due to the 100 ms stride between prediction windows, each instance of a gesture is typically contained in 4 to 6 consecutive windows. Therefore, we set $n = 6$. Including this step, the microcontroller executes the entire prediction pipeline well within the time budget of 100 ms (see Figure 5).

Component	Power Consumption
Microcontroller (MKR1000)	66 mW
IMU sensor (MPU6050)	19.8 mW
BLE (sleep mode)	1.98 mW
BLE (transmission mode)	29.67 mW

Table 1: Power consumption of each component of the GesturePod.

GesturePod: Hardware Evaluation

As discussed before, our goal is to design a robust gesture-recognition device that incurs low cost, low weight, and low power consumption. The GesturePod achieves this goal. First, the GesturePod costs around 10 USD, which is significantly lower than the cost of other accessibility devices (e.g., smartwatch). Second, GesturePod weighs just 49 g. None of the users in our studies had any complaints about the weight of the pod. Having said that, the weight of our design can be further reduced through a tighter integration of the individual components. Third, Table 1 shows the power consumption of each component of the pod. Assuming that the user performs a gesture once every 100 ms, the pod lasts over 21 hours on a single charge. In all our studies, all the users reported that the battery lasted an entire day. Finally, results from our user studies show that the GesturePod robustly and accurately recognizes gestures performed by a number of users.

Android App

We conducted exploratory interviews with 15 volunteers with VI to understand the set of smartphone tasks that can benefit from access via gestures (these users did not participate in our future studies). Users wanted gestures to trigger core tasks such as reading current time and location, responding to phone calls, reading notifications, and integration with apps like WhatsApp, Uber, Maps, etc. (We provide more details of these interviews in the supplementary material.) As our goal with this initial study is to understand if cane gestures are a good mode of interaction, we avoided working with third-party applications and focused on the core tasks.

We mapped a set of core tasks to a simple state machine. We mapped gesture to state transitions to simulate a typical navigation flow in Android. For instance, left twists and right twists are used to scroll forward and backward through the app menu or notifications list, and the user can double tap at any time to return to the home state, much like a home button on android phones. Figure 6 shows the state transition diagram. We designed an Android app to implement this state machine. The app gives users voice feedback through their phone's speakers of earphones. We observed that our users with VI tend to always have one of their earphones in their ear, and they preferred voice feedback through earphones.

IN-LAB STUDY

We conducted an in-lab user study to *quantitatively* evaluate the robustness and effectiveness of the GesturePod. Specifically, we conducted experiments to 1) measure how accurately the GesturePod recognizes gestures performed by users, and 2) measure the effect of using the cane mounted with GesturePod (henceforth referred to as I-Cane) on the

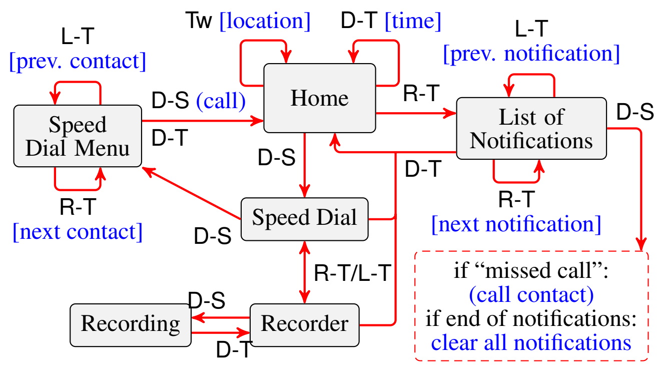


Figure 6: State machine for the app on smartphone. During state transitions, the app explicitly reads out the text corresponding to the item within square brackets.

time taken by users to complete specific tasks on their smartphone. In addition to these measurements, we collected feedback on the usability of the cane on a Likert scale (see supplementary material).

For this study, we recruited 12 users with VI from three Non-Government Organizations (NGOs). All users had received mobility training to use the white cane. Among users, we had diversity in terms of qualifications, gender, age, and handedness. Eight had owned a smartphone for at least three months. Based on consultation with the NGOs and our Institutional Review Board, we compensated each user 7 USD, which is roughly a day's minimum wage for a skilled worker.

Accuracy of GesturePod

All 12 users took part in this experiment. We first trained users on how to perform the five gestures, and then measured the accuracy of GesturePod in recognizing the gestures performed by users. For this experiment, to prevent human bias, we designed an Android app that speaks out a random gesture and waits for the user to perform the gesture with the I-Cane for a period of 10 seconds. We used this app for both training and the subsequent accuracy measurement. Each user brought their own cane for both the in-wild and the in-lab study. The GesturePod was clamped to their cane. In fact, the ability to simply attach the GesturePod to any white cane was one of our key design requirements.

User Training

We first described the project to users. Then, we held their hands and demonstrated how to perform each gesture once. After this initial demonstration, we used our app to train the users where the app requests the user to perform each gesture 5 times. For each attempt, the app notifies the user if GesturePod recognized the gesture. The training lasted 10 minutes for each user.

Accuracy Measurement and Metrics

After the training phase, we use our app to measure the gesture recognition accuracy of GesturePod for each user. The app reads out 25 gestures (5 instances of the 5 gestures) in a *random* order for each user, and tracks if GesturePod correctly recognized the gesture. We measure accuracy for each gesture using two metrics: 1) *recall*, the fraction of times the performed gesture was detected correctly, and 2) *precision*, the fraction of times GesturePod's detected gesture was correct. A high recall is an indicator of low false negatives, and high precision is an indicator of low false positives.

Performed Gesture	Recognized Gesture						
	D-T	R-T	L-T	Tw	D-S	N-G	Recall
D-T	95	0	0	0	0	0	1
R-T	0	86	8	0	0	1	0.91
L-T	0	9	81	0	0	5	0.85
Tw	0	0	0	82	11	2	0.86
D-S	0	0	0	1	93	1	0.98
Precision	1	0.91	0.91	0.99	0.89	-	

Table 2: Confusion matrix for gesture detection performed across all visually impaired users. N-G corresponds to the case when no gesture was recognized.

Results and Discussion

Table 2 presents the results of our experiment. The ij^{th} value in the table is the number of times a gesture in the i^{th} row is detected as a gesture in the j^{th} column. We draw three conclusions. First, GesturePod recognizes double tap (D-T) and double swipe (D-S) with high recall. Second, we observed that while listening to the prompt of gesture to be performed, users sometimes got confused between right and left directionality. For example, on hearing a prompt for right twist they would commit to perform a left twist and then quickly realize and perform right twist. However, in our evaluations, to be fair, our system considered only the first gesture performed after the prompt. Despite this factor, the recall for right twist (R-T) and left twist (L-T) is still around 86%. Third, as a twirl (Tw) performed incorrectly has a signature similar to double swipe (D-S), the model sometimes mispredicts a twirl as a double swipe.

Overall, across all gestures, GesturePod achieves a precision of $92\% \pm 3\%$ (with 95% confidence). Note that our ML model was not trained with data from these users. The fact that we were able to detect gestures even for new-users, highlights the robustness and generalizability of our ML-based gesture recognition model.

Impact on Smartphone Access Time

Through this experiment, we want to analyze the impact of the I-Cane on time taken to complete common activities on the smartphone. 8 users with VI participated in this experiment. All participants from the previous experiment who owned a smartphone for at least three months were enrolled for this experiment. We studied the following five activities.

1. Answer a phone call from a test phone.
2. Call back the last caller from a missed call notification.
3. Start an audio recording and stop.
4. Know the current geographic location.
5. Check for notifications and read out the time.

Based on the feedback from our exploratory interviews, we identified two experimental settings: an *unconstrained setting*, in which one of the user's hands is free and the other hand is holding the cane, and a *constrained setting* in which both of the user's hands are occupied–one hand holding the

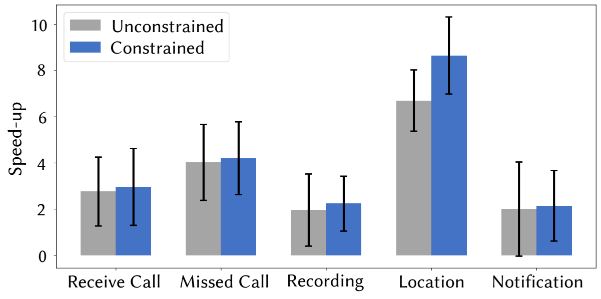


Figure 7: Speedup in task completion times using I-Cane over using smartphone alone.

cane and the other hand holding an item (e.g., a bag). Apart from whether one of the users' hand was free, there were no differences between the two settings.

User Training
At the beginning of this experiment, we read out the state machine (in English/native language) (Figure 6) that maps gestures on the I-Cane to activities on the phone, to each user. We present the exact text used in the supplementary material. The users then practiced the activities using the I-Cane once.

Evaluation
We measure the time taken by each user to complete the five activities in four different scenarios: *constrained* or *unconstrained*, and smartphone-alone or I-Cane. To eliminate "competition effect", we did not a priori inform the participants that we will be timing them. Instead, we video record the experiment and replay the video to measure the time after the study (participants were informed post study). To eliminate "practice effect", across users, we randomize 1) the order of these scenarios, and 2) the order of the activities within each scenario. For the smartphone-alone scenarios, participants used their preferred mode for interacting with the smartphone — such as touch UI, querying voice commands (e.g. "Okay-Google"), assistive app technology (e.g. "EYE-D" Android app), accessibility mode, etc.

Results and Discussion
When using the I-Cane, all the users were able to complete all the activities in both the unconstrained and constrained settings. In contrast, some users were unable to complete some of the activities using the smartphone-alone. Specifically, three users were unable to complete the recording activity in both unconstrained and constrained settings; and one user was unable to identify the location and read out notifications in the constrained setting.

Figure 7 plots the average speed-up in task completion times using I-Cane compared to the smartphone-alone for each of the five activities across all the users, along with the standard deviation. While computing speedup, we exclude data points where a user was unable to complete the activity on the smartphone-alone.

We draw two conclusions from our results. First, in both unconstrained and constrained settings, users were able to complete tasks 2×–9× faster using I-Cane than using their smartphone-alone. Second, as expected, the speedups are

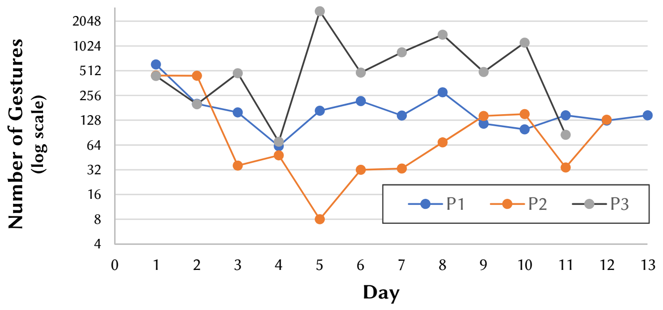


Figure 8: Number of gestures performed by users on the days we detected at least one gesture.

slightly higher in the constrained setting, especially for the location task. For the location task, most users had to navigate multiple screens to identify the location using the smartphone-alone, whereas, with the I-Cane, they could complete the task with a single gesture.

While one user successfully used Google voice commands to complete some tasks, other users stated that they generally do not prefer voice commands for various reasons:

1. "I reside in a hostel with other blind people. If all of us start talking to use the phone, there would be a mess."
2. "It does not recognize vernacular accents."
3. "On the streets, with the honking and noise, it does not recognize our voices."

IN-WILD STUDY
To understand the usefulness and adaptability of GesturePod in the real-world, we conducted a 15 day in-wild user study. We recruited three users with visual impairment from two NGOs. All three users had received mobility training and use a smartphone. One of the users had participated in our in-lab user study.

We briefed each user about GesturePod and trained them to perform the gestures and their effect on their Android phones. Additionally, we trained each user on the operation of GesturePod, switching it on/off, charging it, and connecting it to their phone. We attached GesturePod to each user's cane and connected the pod to their phone. Following this training, we informed the user that, for the remainder of the study, they could interact with their phone either using their normal method or through gestures on the cane, whichever they deemed fit. We did not initiate contact with the users until the end of the study.

Evaluation Metrics
During the study, our app logged all the gestures performed by the user. At the end of the study, we collected feedback from each user in three ways: 1) System Usability Scale (SUS) to measure usability of the system, 2) Likert scale responses on the design of the system, and 3) semi-structured interviews. We recorded, transcribed, and analyzed the interviews to identify emergent themes. To prevent response biases, an independent individual not part of our project collected the SUS and Likert scale responses.

Quantitative Results

Figure 8 plots the number of gestures we detected for each of the three users (P1—P3). As there was a high variance in the number of gestures across users (possible due to differing phone usage), the figure plots the number of gestures in log scale. In addition, during our interviews, we learned that the users did not use their cane on days they were with their family members due to social stigma (see next section). Therefore, we plot the number of gestures for each user only for days on which we detected at least one gesture (Break-up of gestures per user per day provided in supplementary material). Our results show that each user used the cane on at least 11 days out of 15 days of study, and each user's median number of gestures performed was more than 58. We noticed that P3 performed a large number of gestures on day 5. During our interviews, he told us that he had demonstrated the cane to his students on that day.

As our app cannot distinguish between a real (intended) gesture and a false positive, to track false positives — we asked users if they experienced any false positive during their cane use. Two of the users mentioned occurrences of few false positives across all 15 days: P1 (4 false double taps) and P2 (20 false double swipes). These numbers are small in comparison to the total number of gestures performed by the users. More importantly, the users did not report significant inconvenience due to false positives.

Users gave positive feedback on the System Usability Scale: P1–90, P2–75, and P3–95. The responses for the Likert scale were also encouraging. We provide the exact questions for SUS and the anchors for the Likert scale in the supplementary material. Table 3 shows the responses for the Likert scale.

These results indicate that our approach of using gestures on the cane to interact with the smartphone can be robust, promising, and potentially useful in the real world.

Qualitative Feedback

We conducted semi-structured interviews with users at the end of the study. In general, the users found I-Cane to be most useful in constrained scenarios. In contrast, it was least useful when the users were with their family members, due to limited requirement of independent navigation as well as due to social stigma. All the users mentioned that they did not feel any strain while performing the gestures, and the pod (and its weight) did not effect their normal cane usage.

Constrained Scenarios

All users reported I-Cane to be helpful in constrained scenarios, i.e., when both their hands were occupied. One such scenario is during the use of public transit, where seating is often unavailable. A common challenge in such situations is having to balance oneself by holding on to a railing. The exact moment of a cellphone vibrating can be jarring and distracting, and trying to remove it from one's pocket or bag and operate it can cause tripping and injury.

> *When I was traveling by metro, holding the cane in one hand and the railing (support) in the other hand. I am getting calls, before it was not possible (to answer them). Now I can talk using the cane.* —P2

Question	P1	P2	P3
Was performing the gestures comfortable?	3	5	5
Were the gestures intuitive to perform?	5	5	5
Was the cane able to detect your gestures accurately?	3	4	4
How easy did you find it to remember the gestures?	5	5	5
How much effort was required to get used to the way the gestures are to be performed?	5	4	5
How would you rate ease to use and operate the app?	4	5	4
How would you rate the physical design of the pod?	5	5	5
How would you rate the overall product experience?	3	4	5

Table 3: User ratings from in-wild study on different aspects of GesturePod. Rating: 1 - lowest, 5 - highest.

Another common situation that people report is needing to know the current time or location, especially when they are travelling. Here too, one needs both hands, if touching a tactile watch, or managing multiple swipes or strokes on a touchscreen to get the time or location.

> *When I am traveling in bus, we are standing and it is easy to know the time and location using the cane.* —P3

Even though we did not intend the I-cane to be used in motion, users mentioned that they can perform double-tap, right-twist and left-twist while in motion. This can potentially help them attend to time-critical tasks.

> *Earlier I had to take my phone from pocket and stand somewhere to pick up a call, but now I can answer it while walking.* —P3

One disadvantage with not having a visual interface to a smartphone is the time taken to react to a call due to the inability to glance at a ringing device. Moreover, when the phones need to be kept in a secure location, such as locked in a purse or in a secure pocket area, the reaction time in removing the device can be long enough to delay a timely interaction. All users noted the benefit of I-Cane in being able to complete tasks without needing to physically handle a phone and deal with the touchscreen.

> *Many times I would miss calls due to delay in accepting them. Now that does not exist.* —P2

Users found the cane to be useful even when one of their hands were free.

> *In footpaths, even without any luggage in our hand, we have to stop and then remove the phone from our pockets, and then note the time. This makes people behind us to stop. Also, this is very long. With cane, we can know the time immediately.* —P1

Situational Awareness

Locational awareness is a critical part of independent life. Our app logs indicate that all users frequently used the cane to query for location. In fact, all users explicitly mentioned liking the ability to query their location with a single gesture.

In unknown locations, sometimes people won't tell us correctly where we are, sometimes it's hard to find people. With the cane, we can now know the location on our own. Before taking the phone out would take time. —P1

Locational awareness may require reassurance or repeated checking within very short intervals. This is common for sighted people using mapping technologies (e.g., Google maps) in which one may glance at a screen several times as they move. Such repeated queries for location can be particularly tedious for people with VI just using their smartphone UI. With I-Cane, the users were able to locate themselves quickly, and repeatedly.

If I want to go from office to Parangipalya (neighborhood), sometimes I will get confused in the cross-streets... if I check the location it was saying intersection of 27th main and 18th cross. So I was able to easily find out I am in this particular cross and I can navigate well. —P3

Cultural Conditioning

We found that the notion of people with VI navigating on their own was still alien. This was true for users whose family members would often accompany them to various locations rather than allow them to travel on their own. While this was internalized as care on part of family members to ensure that they were not hurt, the net outcome was limited experience with outdoor spaces.

Users also referred to the use of the cane as stigmatizing because they identified one as being disabled. Consequently, there were entire days during which some of our users did not use the cane. As two users put it,

I am a totally blind person and I cannot live without the cane ... yet when I am at home, my parents do not let me use the cane. —P3

Stigma was related not just to the individual, but also to those around them by extension, since disability was sometimes seen as something embarrassing and therefore to be hidden from public view.

My sister and mother do not allow me to use the cane when they are around me, as I was getting engaged. –P2

Such issues of stigma are relevant to design decisions, since on one hand, there is a need for greater social awareness of disability as diversity, but there is also a need to consider the possibility that people with VI may find themselves forced to use devices or aids that do not reveal their disability to those around them.

DISCUSSION

In this section, we summarize insights that we gained during our study. We hope that these insights will guide future research that aims to design solution for a similar context.

Safety of Gestures

Among the five gestures, we felt that the double-tap gesture (tapping the cane to the ground twice) may create noise that may either disturb people in the vicinity or draw attention to the user. However, users mentioned that this was not a concern. On the other hand, one user mentioned that a "bigger" twirl could potentially hit someone or something in the immediate vicinity of the user. Given that these two gestures were the most popular among users, it seems like users adapted to performing these gestures in a safe and polite manner.

Better Integration with phone OS

One of the user wished that he could "operate all of the mobile" using gestures. This is possible by combining gestures with the native android accessibility app, i.e., TalkBack. But we believe that easier and faster access to certain frequent and time-critical tasks on smartphone might be a better application of GesturePod; we will study both these approaches in future work.

Additional Functionality

Android's Google assistant requires an always-on microphone, which can drain the battery rapidly on low-end phones. In fact, one user mentioned that he would like one gesture to wake up the Google assistant. Similarly, another user wanted a mechanism to know the battery level of the GesturePod, so that they can charge the pod at an appropriate time. Our apps state machine can be extended to include these functionalities. Similarly, some users requested gestures to be mapped to an action of their choice (e.g., call a specific person). These requests suggest that a) users should be able to customize our Android app's state machine, and b) users found the alternate functionality organization, that our cane enables, to be of great advantage.

What about buttons?

Our goal is to determine if gesture detection on white cane is technically feasible and practical. Our goal is not to conclude that gestures are the best mode of interaction for cane users. We suspect that creating a button-based cane in itself requires significant design exploration: how many buttons, location of buttons, mapping between buttons and actions, etc. Once deployed, unlike a software based gesture recognition system, a hardware based button system is not extensible. Further, to avoid false triggers, buttons have to be placed away from the place where the users hold the cane. This can be clumsy for users with VI and require them to locate/identify each button before performing any action (e.g., imagine different orientations of the cane). This challenge with buttons was highlighted by users with VI in our interviews. However, note that gestures are complementary to buttons, and it is quite likely that a final solution may marry gestures with buttons.

CONCLUSION

We introduced GesturePod, an easy-to-use gesture-recognition device that can be clamped on any white cane so that gestures of the cane can be used to access a smartphone. GesturePod is a touch free, low-cost (less than 10 USD) plug-and-play device, that has a short learning curve, and does not require carrying around additional devices other than the cane. GesturePod's real-time and power-efficient gesture recognition is enabled by a carefully engineered ML pipeline that runs entirely on a tiny micro-controller housed inside the pod. Our in-lab and in-wild user studies suggested that 1) GesturePod can robustly recognize gestures across the wide range of environments recorded by our users, and 2) GesturePod could potentially improve access to smartphone for specific tasks. Both our results and qualitative feedback from the users suggest that, for persons with VI, performing gestures on the cane is a promising mode of interaction with their smartphone, especially in workplace or outdoors.

ACKNOWLEDGEMENTS

We are grateful to Edward Cutrell, Jacki O'Neill and Manohar Swaminathan for helping us with the design and protocols for user studies; Joyojeet Pal and William Thies for helpful feedback on the paper; Abhishek Vijaya Kumar and Rajanikant Deshmukh for designing the Android App; Aditya Kusupati for many helpful discussions.

REFERENCES

[1] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2043–2046. DOI: http://dx.doi.org/10.1145/1978942.1979238

[2] Daniel Ashbrook, Carlos Tejada, Dhwanit Mehta, Anthony Jiminez, Goudam Muralitharam, Sangeeta Gajendra, and Ross Tallents. 2016. Bitey: An Exploration of Tooth Click Gestures for Hands-free User Interface Control. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 158–169. DOI: http://dx.doi.org/10.1145/2935334.2935389

[3] Lawrence K. Au, Winston H. Wu, Maxim A. Batalin, Thanos Stathopoulos, and William J. Kaiser. 2008. Demonstration of Active Guidance with SmartCane. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN '08)*. IEEE Computer Society, Washington, DC, USA, 537–538. DOI:http://dx.doi.org/10.1109/IPSN.2008.52

[4] Jared M. Batterman, Vincent F. Martin, Derek Yeung, and Bruce N. Walker. 2018. Connected cane: Tactile button input for controlling gestures of iOS voiceover embedded in a white cane. *Assistive Technology* 30, 2 (2018), 91–99. DOI: http://dx.doi.org/10.1080/10400435.2016.1265024

[5] Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 1901–1910.

[6] Don Kurian Dennis, Shishir G. Patil, Chirag Pabbaraju, Nadeem Shaheer, Harsha Simhadri, Vivek Seshadri, Manik Varma, and Prateek Jain. 2018. *GesturePod: Gesture-based Interaction Cane for People with Visual Impairments*. Technical Report MSR-TR-2018-14. Microsoft.

[7] J. Faria, S. Lopes, H. Fernandes, P. Martins, and J. Barroso. 2010. Electronic white cane for blind people navigation assistance. In *2010 World Automation Congress*. 1–7.

[8] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M. Cardoso. 2010. Preprocessing Techniques for Context Recognition from Accelerometer Data. *Personal Ubiquitous Comput.* 14, 7 (Oct. 2010), 645–662. DOI: http://dx.doi.org/10.1007/s00779-010-0293-9

[9] Fitbit. 2007. Fitbit. (2007). https://www.fitbit.com/com/home.

[10] German H. Flores and Roberto Manduchi. 2016. WeAllWalk: An Annotated Data Set of Inertial Sensor Time Series from Blind Walkers. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. ACM, New York, NY, USA, 141–150. DOI: http://dx.doi.org/10.1145/2982142.2982179

[11] A. J. Fukasawa and K. Magatani. 2012. A navigation system for the visually impaired an intelligent white cane. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 4760–4763. DOI: http://dx.doi.org/10.1109/EMBC.2012.6347031

[12] Maribeth Gandy, Thad Starner, Jake Auxier, and Daniel Ashbrook. 2000. The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring. In *Proceedings of the 4th IEEE International Symposium on Wearable Computers (ISWC '00)*. IEEE Computer Society, Washington, DC, USA. http://dl.acm.org/citation.cfm?id=851037.856538

[13] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma, and Prateek Jain. 2017. ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 1331–1340. http://proceedings.mlr.press/v70/gupta17a.html

[14] Björn Hartmann, Leith Abdulla, Manas Mittal, and Scott R. Klemmer. 2007. Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 145–154. DOI: http://dx.doi.org/10.1145/1240624.1240646

[15] Yongsik Jin, Jonghong Kim, Bumhwi Kim, Rammohan Mallipeddi, and Minho Lee. 2015. Smart Cane: Face Recognition System for Blind. In *Proceedings of the 3rd International Conference on Human-Agent Interaction (HAI '15)*. ACM, New York, NY, USA, 145–148. DOI: http://dx.doi.org/10.1145/2814940.2814952

[16] Lei Jing, Yinghui Zhou, Zixue Cheng, and Tongjun Huang. 2012. Magic Ring: A Finger-Worn Device for Multiple Appliances Control Using Static Finger Gestures. *Sensors* 12, 5 (2012), 5775–5790. DOI: http://dx.doi.org/10.3390/s120505775

[17] Jin Sun Ju, Eunjeong Ko, and Eun Yi Kim. 2009. EYECane: Navigating with Camera Embedded White Cane for Visually Impaired Person. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '09)*. ACM, New York, NY, USA, 237–238. DOI: http://dx.doi.org/10.1145/1639642.1639693

[18] Sung Yeon Kim and Kwangsu Cho. 2007. Usability and design guidelines of smart canes for users with visual impairments. *International Journal of Design* 7, 1 (2007), 99–110. http://www.ijdesign.org/index.php/IJDesign/article/view/1209/559

[19] Louis Kratz, Daniel Morris, and T. Scott Saponas. 2012. Making Gestural Input from Arm-worn Inertial Sensors More Practical. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1747–1750. DOI:http://dx.doi.org/10.1145/2207676.2208304

[20] Sven Kratz and Maribeth Back. 2015. Towards Accurate Automatic Segmentation of IMU-Tracked Motion Gestures. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 1337–1342. DOI: http://dx.doi.org/10.1145/2702613.2732922

[21] Sven Kratz, Michael Rohs, and Georg Essl. 2013. Combining Acceleration and Gyroscope Data for Motion Gesture Recognition Using Classifiers with Dimensionality Constraints. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*. ACM, New York, NY, USA, 173–178. DOI: http://dx.doi.org/10.1145/2449396.2449419

[22] Ravi Kuber, Amanda Hastings, Matthew Tretter, and Dónal Fitzpatrick. 2012. Determining the accessibility of mobile screen readers for blind users. (2012).

[23] Fingertips Lab. 2017. O6. (2017). https://www.kickstarter.com/projects/55699542/o6-free-your-eyes.

[24] Walter S. Lasecki, Young Chol Song, Henry Kautz, and Jeffrey P. Bigham. 2013. Real-time Crowd Labeling for Deployable Activity Recognition. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1203–1212. DOI: http://dx.doi.org/10.1145/2441776.2441912

[25] Je Seok Lee, Heeryung Choi, and Joonhwan Lee. 2015. TalkingCane: Designing Interactive White Cane for Visually Impaired People's Bus Usage. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 668–673. DOI: http://dx.doi.org/10.1145/2786567.2793686

[26] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. In *2009 IEEE International Conference on Pervasive Computing and Communications*. 1–9. DOI: http://dx.doi.org/10.1109/PERCOM.2009.4912759

[27] Zhiyuan Lu, Xiang Chen, Zhangyan Zhao, and Kongqiao Wang. 2011. A Prototype of Gesture-based Interface. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 33–36. DOI: http://dx.doi.org/10.1145/2037373.2037380

[28] David Mace, Wei Gao, and Ayse Coskun. 2013. Accelerometer-based Hand Gesture Recognition Using Feature Weighted Naive Bayesian Classifiers and Dynamic Time Warping. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion (IUI '13 Companion)*. ACM, New York, NY, USA, 83–84. DOI: http://dx.doi.org/10.1145/2451176.2451211

[29] Joseph Malloch, Carla F. Griggio, Joanna McGrenere, and Wendy E. Mackay. 2017. Fieldward and Pathward: Dynamic Guides for Defining Your Own Gestures. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4266–4277. DOI: http://dx.doi.org/10.1145/3025453.3025764

[30] Jani Mäntyjärvi, Juha Kela, Panu Korpipää, and Sanna Kallio. 2004. Enabling Fast and Effortless Customisation in Accelerometer Based Gesture Interaction. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM '04)*. ACM, New York, NY, USA, 25–31. DOI: http://dx.doi.org/10.1145/1052380.1052385

[31] David A. Mellis, Ben Zhang, Audrey Leung, and Björn Hartmann. 2017. Machine Learning for Makers: Interactive Sensor Data Classification Based on Augmented Code Examples. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. ACM, New York, NY, USA, 1213–1225. DOI:http://dx.doi.org/10.1145/3064663.3064735

[32] Microsoft. 2017a. Microsoft Soundscape. (2017). https://www.microsoft.com/en-us/research/product/soundscape/.

[33] Microsoft. 2017b. Seeing AI. (2017). https://www.microsoft.com/en-us/seeing-ai/.

[34] Annika Muehlbradt, Varsha Koushik, and Shaun K. Kane. 2017. Goby: A Wearable Swimming Aid for Blind Athletes. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. ACM, New York, NY, USA, 377–378. DOI: http://dx.doi.org/10.1145/3132525.3134822

[35] Uran Oh, Lee Stearns, Alisha Pradhan, Jon E. Froehlich, and Leah Findlater. 2017. Investigating Microinteractions for People with Visual Impairments and the Potential Role of On-Body Interaction. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. ACM, New York, NY, USA, 22–31. DOI: http://dx.doi.org/10.1145/3132525.3132536

[36] Philip O'Keefe. 2007. People with disabilities in India: from commitments to outcomes (English). *Washington, DC: World Bank* 1, 41585 (2007), 1–185.

[37] Tomàs Pallejà, Marcel Tresanchez, Mercè Teixidá, and Jordi Palacin. 2010. Bioinspired Electronic White Cane Implementation Based on a LIDAR, a Tri-Axial Accelerometer and a Tactile Belt. *Sensors* 10, 12 (2010), 11322–11339. DOI: http://dx.doi.org/10.3390/s101211322

[38] Sumita Sharma, Saurabh Srivastava, Krishnaveni Achary, Blessin Varkey, Tomi Heimonen, Jaakko Hakulinen, Markku Turunen, and Nitendra Rajput. 2016. Gesture-based Interaction for Individuals with Developmental Disabilities in India. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. ACM, New York, NY, USA, 61–70. DOI: http://dx.doi.org/10.1145/2982142.2982166

[39] Vaibhav Singh, Rohan Paul, Dheeraj Mehra, Anurag Gupta, Vasu Dev Sharma, Saumya Jain, Chinmay Agarwal, Ankush Garg, Sandeep Singh Gujral, M. Balakrishnan, Kolin Paul, P.V.M. Rao, and Dipendra Manocha. 2010. Smart cane for the visually impaired: Design and controlled field testing of an affordable obstacle detection system. In *12th International Conference on Mobility and Transport for Elderly and Disabled Persons (TRANSED '10)*.

[40] Dring Alert System. 2017. The Connected Cane. (2017). http://dring.io/en/the-connected-cane/.

[41] APH Tech. 2017. Long Cane Techniques. (2017). https://tech.aph.org/sbs/04_sbs_lc_study.html.

[42] GingerMind Technologies. 2017. Eye-d. (2017). https://www.eye-d.in/.

[43] T Warren Liao. 2005. Clustering of time series data-a survey. *Pattern Recognition* 38, 11 (2005), 1857–1874.

[44] WeWALK. 2018. WeWALK SMART CANE. (2018). https://get.wewalk.io/.

[45] WhiteCaneDay 2017. White Cane Day. (2017). http://www.whitecaneday.org/canes/.

[46] Michele A. Williams, Caroline Galbraith, Shaun K. Kane, and Amy Hurst. 2014. "Just Let the Cane Hit It": How the Blind and Sighted See Navigation Differently. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. ACM, New York, NY, USA, 217–224. DOI:http://dx.doi.org/10.1145/2661334.2661380

[47] Koji Yatani and Khai N. Truong. 2012. BodyScope: A Wearable Acoustic Sensor for Activity Recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. ACM, New York, NY, USA, 341–350. DOI: http://dx.doi.org/10.1145/2370216.2370269

[48] Hanlu Ye, Meethu Malu, Uran Oh, and Leah Findlater. 2014. Current and Future Mobile and Wearable Device Use by People with Visual Impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3123–3132. DOI: http://dx.doi.org/10.1145/2556288.2557085

[49] Ying Yin and Randall Davis. 2013. Gesture Spotting and Recognition Using Salience Detection and Concatenated Hidden Markov Models. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction (ICMI '13)*. ACM, New York, NY, USA, 489–494. DOI: http://dx.doi.org/10.1145/2522848.2532588

[50] Yuhang Zhao, Cynthia L. Bennett, Hrvoje Benko, Edward Cutrell, Christian Holz, Meredith Ringel Morris, and Mike Sinclair. 2018a. Demonstration of Enabling People with Visual Impairments to Navigate Virtual Reality with a Haptic and Auditory Cane Simulation. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Article D409, 4 pages. DOI: http://dx.doi.org/10.1145/3170427.3186485

[51] Yuhang Zhao, Cynthia L. Bennett, Hrvoje Benko, Edward Cutrell, Christian Holz, Meredith Ringel Morris, and Mike Sinclair. 2018b. Enabling People with Visual Impairments to Navigate Virtual Reality with a Haptic and Auditory Cane Simulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 116, 14 pages. DOI: http://dx.doi.org/10.1145/3173574.3173690