MobiReader: A Wearable, Assistive Smartphone Peripheral for Reading Text

by

Marcelo Rene Polanco II

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Masters of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

© Massachusetts Institute of Technology 2015. All rights reserved.

Certified by.....

Pattie Maes Professor of Media Arts and Sciences Thesis Supervisor

Accepted by

Professor Albert R. Meyer Chairman, Masters of Engineering Thesis Committee

MobiReader: A Wearable, Assistive Smartphone Peripheral for Reading Text

by

Marcelo Rene Polanco II

Submitted to the Department of Electrical Engineering and Computer Science on May 22, 2015, in partial fulfillment of the requirements for the degree of Masters of Engineering in Computer Science and Engineering

Abstract

Accessing printed text in mobile situations is inconvenient for people with visual impairments. Existing smartphone technologies often require refined aiming and focusing actions that prove difficult for the target user group to perform. In an attempt to mitigate these issues, the FingerReader was developed to assist in reading printed text. This thesis discusses the evolution of the FingerReader to the MobiReader, a mobile version of its predecessor meant to extend its potential as a pocketable, assistive reading device. The remainder of this document discusses the implementation and technical components of the MobiReader for the Android platform and reports its evaluation in a user study.

Thesis Supervisor: Pattie Maes Title: Professor of Media Arts and Sciences

Acknowledgments

I'd like to thank my family and Sara Lowe for supporting me thoroughly throughout this endeavor. I'd also like to thank Roy Shilkrot and Michael Chang for their guidance and encouragement and Pattie Maes for imparting this valuable opportunity to learn.

Contents

1	Intr	roduction					
	1.1	1 Motivations for MobiReader					
	1.1.1 Applications for those without visual impairments						
	1.2	Relate	ed Work	16			
		1.2.1	FingerReader and EyeRing	16			
		1.2.2	Mobile Reading Assistive Technologies	17			
		1.2.3	Device-based Assistive Technologies	18			
	1.3	Contri	butions of this thesis	20			
2	Imp	lemen	tation	21			
	2.1	The MobiReader System					
		2.1.1	MobiReader hardware	21			
		2.1.2 Android application					
	2.2	Desigr	1	23			
		2.2.1	User Experience	23			
	2.3	Finger	Reader on the Android Platform	26			
	2.4	Archit	ecture	26			
		2.4.1	High Level System Construction	27			
		2.4.2	Front-End Description	27			
		2.4.3	Backend Description	29			
		2.4.4	Computer vision algorithm	30			
		2.4.5 Thread Level Description					

3	Eva	luatio	n	35			
	3.1	1 Goals of the user tests					
	3.2	Proce	dure	36			
		3.2.1	Setup	36			
		3.2.2	Pre-Usage Questionnaire	38			
		3.2.3 Practice					
		3.2.4	Usage Tasks	39			
		3.2.5	Post-Usage Questionnaire	42			
	3.3	Partic	eipants	42			
	3.4	Analy	sis	43			
		3.4.1	Data synchronisation and alignment	43			
		3.4.2	Extracted measurements	45			
	3.5	Result	ts	46			
		3.5.1	Quantitative measurements	46			
		3.5.2	Qualitative feedback	48			
	3.6	3.6 Discussion					
	3.7	Futur	e Work	51			
4	Cor	nclusio	n	55			
	4.1	Lesso	ns Learned	55			
		4.1.1	Further Usage	55			
		Further Testing	56				
		4.1.3	Form Factor	57			
4.2 Android Discussion		Andro	oid Discussion	59			
		4.2.1	Conquering the native environment	59			
		4.2.2	Limitations	60			
		4.2.3	Future Iterations	61			
	4.3 Conclusion						

List of Figures

1-1	The MobiReader camera peripheral	14
2-1	A screenshot of the MobiReader interface during usage	22
2-2	High level diagram of the MobiReader application architecture. Blue	
	components represent front-end interfaces, red represents back-end com-	
	ponents, and green signifies messages passed in the form of Android	
	Intents	27
2-3	(a) the front-end interface featuring the camera view of the MobiReader	
	peripheral. (b) the preferences interface where a user may customize	
	the application settings	28
2-4	The state diagram from $[16]$ for the computer vision algorithm modified	
	to include the new/improved MobiReader states (marked in red)	31
2-5	A depiction of the active threads running within the application. Na-	
	tive threads are depicted in red while threads in the Java environment	
	are depicted in black. Black arrows signify message passing through	
	shared memory. The green arrow signifies message passing through	
	Android Intents. Blue dashed arrows signify the initiator of the thread.	33
3-1	Testing environment with P7 in midst of reading with the MobiReader	37
3-2	Display of the iPad testing interface used to track the users finger	
	position. Pages were aligned with the grey boundary and taped to	
	tablet to secure the position. As users trace their finger over their	
	page in the grey boundary, their finger position is recorded \ldots	38
3-3	The practice sheet used during testing	39

3-4	Aligned data from the iPad of a single session for P4, a completely blind					
	person. The green-blue markings are a visualization of the <i>Distance</i>					
	feedback the user received, while the red markings are when words were					
	spoken out loud. The middle line of the text was missed, however the					
	user was still able to recover 73% of the words in total and spent over					
	51% of the time line-tracking	44				
3-5	Individual success in reading per participant. Note: we were unable to					
	extract timing data for P2	46				
3-6	Reading success as crossed by (a) feedback type (a), and (b) font size.					
3-7	Qualitative measurement of perceived success. 1 - best, 5 - worst. $\ .$.					
3-8	Qualitative measurement of independence. 1 - best, 5 - worst. \ldots .	49				
3-9	Qualitative measurement of understanding the audio feedback. $1\ -$					
	best, 5 - worst.	50				

List of Tables

3.1	Pre-usage questions.					
3.2	Usage tasks and their descriptions					
3.3	Task all otment for participants where ${\cal D}$ signifies a task with distance					
	feedback and $D+A$ signifies a test with distance and angle feedback.					
	Participant 0 is excluded from this table. \ldots \ldots \ldots \ldots \ldots \ldots					
3.4	Post-usage questions asked after completion of all tasks					
3.5	Details of the participants in our user study. Note: P0 was a pilot					
	participant.	43				

Chapter 1

Introduction

Accessing printed reading material in an unstructured or unfamiliar environment is still a major challenge for people with visual impairments (VI). To bridge this gap, researchers in academia and industry proposed solutions, such as Eye-Pal¹ or the OrCam². However these proposed solutions are difficult to operate or are not necessarily readily available. In interviews conducted in 2014 by Shilkrot et al. it was revealed that individuals with visual impairments had more difficulty focusing the camera on material, aligning any camera aids with the text and using assistive technology in everyday settings. These areas of improvement were also reiterated through the needs and problems expressed by interviewees included in this study and appeal to the necessity for text-access assistive technology that can overcome the hurdles of lighting, focus, aim and environment.

A variety of issues both physical and social inhibit individuals with visual impairments from reading printed text. An example of this is Braille. Braille, while it exists as a social standard, only exists translated from 1% of written texts [22] and it is reported that less than 10% of the population with visual impairments has extensive training with Braille and would be considered fluent [1]. In addition, the varying level of impairments that an individual may be affected by lead to different requirements of assistive devices. Already existing assistive devices such as portable scanners are

¹http://www.abisee.com/products/eye-pal.html

²http://www.orcam.com/



Figure 1-1: The MobiReader camera peripheral

often large and lack mobility, providing a burden to the user.

The FingerReader [16] is an assistive device that reads printed text to individuals with visual impartments through an interactive reading experience. The user scans their finger along the line of text they wish to read and provides the FingerReader with the information they wish to receive audibly. The experience provides an action for the user that is a direct parallel to the actions performed in reading Braille.

The MobiReader, the next iteration of the FingerReader system, aims to provide a solution not offered by traditional solutions by providing a truly mobile visual impairment assistive device in the form of a wearable, form fitting ring outfitted with a small camera. This thesis aims to convey the motivation and technical advances behind the transition from the original FingerReader, a device paired with a computer, to the MobiReader, a version of the FingerReader that operates on a mobile phone.

1.1 Motivations for MobiReader

Despite the number of available solutions, reading printed material remains a problem for visually impaired users, especially in unfamiliar environments. Many solutions contain a number of usability problems that make their operation as much of a struggle as it is a convenience.

Visually impaired users have noted the importance of efficiency in their daily interactions with personal devices, however, current mobile technologies present as much challenge as benefit [10]. In contrast to standard smartphones apps that offer text reading by use of the back-facing camera, the MobiReader operates as a peripheral camera smartphone extension worn on the user's finger to allow for finer control. When paired with the accompanied software, the system reads printed text aloud to the user while providing continuous, real-time audio feedback about the user's finger placement relative to a line of text. Using a standard smartphone is key, since these devices are both prolific within the VI persons community and have ample computation power in recent generations. A peripheral device, which could be made cheaper as it uses less components, can spare the user from purchasing a costly specialized device or even a new smartphone by simply adding external capabilities. A recent survey reveals that peripheral and complementary devices are welcome in the VI community [25], as Bluetooth-coupled headsets and braille displays and keyboards are in wide use.

1.1.1 Applications for those without visual impairments

By the nature of the MobiReader's construction, one of the most motivating prospects of the device is its potential for further use cases beyond solely reading textual content. The potential for its use as a teaching tool is naturally derived from its use of the prelinguistic pointing motion that has become associated with the development of children's linguistic skills [21]. Just as the device is used to read content for the purpose of discovering the meaning of that content, the device could be used with nonvisually impaired audiences to teach a user how to read. Being that the device already builds upon computer-vision algorithms to convey the primary experience, there is potential to extend these algorithms to perform object recognition, further extending the potential use cases of this device. With these potential use cases in mind, it's important that the device be accessible and portable, motivating the necessity for the mobility this project aims to provide.

1.2 Related Work

Much of the background on assistive finger worn camera devices is adequately covered in related prior work [14, 19, 16]. This section will review interesting work in the related fields.

1.2.1 FingerReader and EyeRing

Early prototypes of the MobiReader were presented in [14, 16]. These devices presented a camera mounted on a ring that the user would wear on their index finger. The devices, when tethered to a computer via USB, stream a direct view of what the user points to. Using computer vision algorithms, a user's fingertip is detected and serves as an input. The tip of the finger provides information as to what line is being read and what word the user desires to read next. The experience the rings present to their users amounts to one in which a user would simply point and scan their finger along printed text and the computer would read the words found. As the user scans their finger over printed text, image-processing algorithms running on paired computation hardware interpret words. The interpreted text is then read out to the user using text to speech libraries. Should the user veer off a line of text or meet its end, the software gives feedback to the user about their finger placement and helps them appropriate their finger position. These rings present users with an interactive experience similar to that one would have when reading braille.

Before this project, the FingerReader required being tethered to a computer so that the computer vision processing could adequately perform. This was done to provide a strong computation source that could handle the image processing quickly to keep up with the user's reading pace, but it severely limited the variety of settings that the device could be used in. The onset of faster smartphones and their prolific presence has made it both possible and favorable for the original FingerReader algorithms to operate on a mobile platform.

1.2.2 Mobile Reading Assistive Technologies

Application stores on all major mobile platforms provide a variety of accessibility applications that extend the functionality of their associated devices. Many applications, such as kNFB Reader³, LookTel⁴, and Prizmo⁵, operate by directing the user to take a picture of text with the backfacing camera. Once a picture has been obtained, the textual content is interpreted with an OCR algorithm and read back to the user through a text to speech interface.

A variety of limitations exist with this particular usage model that provide opportunity for improvement. The primary and most complicated step for the user with a visual impairment requires that they align and focus their camera on the desired textual area without the knowledge of what the camera is taking a picture of. Once the user has accomplished this, another hurdle must be overcome as the user attempts to navigate the text for desired content, which, depending on the application, may or may not be an available option. Finally, the accuracy of the OCR may lead to error, especially if the text is formatted inconsistently or if the environmental light conditions lead to an unfavorable picture.

Another approach mobile solutions have explored utilizes crowd sourcing and/or peer-to-peer assistance to aid in all visual tasks. VizWiz [3], one such application that applies this approach, asks that a user take a picture and ask a question about the associated picture. Through crowdsourcing, an answer to the question is provided. BeMyEyes⁶ is another, similar application that operates in the same way but enables the user to begin a live chat through which their question can be answered through

³http://www.knfbreader.com/

⁴http://www.looktel.com/

⁵http://www.creaceed.com/prizmo

⁶http://www.bemyeyes.org/

a video stream.

With these solutions, aiming and focusing the camera is still a requirement, but the requirements on the images provide more tolerance for error. A variety of other limitations are introduced with the requirement for network connectivity and its natural consequence of introducing latency into the user experience. Rather than instantly receiving feedback, a user must wait for their content to upload and then must wait for a candidate to answer the question. Any inaccuracies in the response cannot be verified easily.

1.2.3 Device-based Assistive Technologies

While a multitude of software applications exist in smartphone marketplaces to extend those devices functionalities, alternative solutions have been explored using the realm of external hardware to aid with visual impairments.

The use of tools to aid with visual impairments can be traced to the Optaphone introduced in 1914 [4]. Many articles cite a more recent spinoff, the Optacon, as one of the best in recent decades [20]. Those who had the chance to use it before its discontinuation, within the group of participants in the evaluation chapter of this thesis, confirmed its dominant experience.

The most relevant works in academia were already mentioned in [16], such as Kane et al.'s AccessLens [9] and Yi and Tian's camera based solution [26]. Other works not involving computer vision, such as El-Glaly's finger-reading iPad [6] and Yarrington's skimming algorithm [24], demonstrate the need to create an equilibrium between visual and non-visual readers by importing aspects of visual reading to assistive technology for VI persons.

Finger Worn Devices

A variety of other smart wearable devices have surfaced in recent years that often capitalize on the prominence of mobile smart phones to extend their functionality and create more readily available experiences. Product releases from industry titans such as the Apple Watch⁷ from Apple and Android Wear⁸ from Google validate this trend.

In particular, a number of smart finger-worn devices have entered the market that extend the functionality of smart phones. In the Digital Digits article [17], Shilkrot et al. identify over 140 instances of smart finger-worn devices and note that the rate of their inception is steadily increasing.

The area of finger worn camera devices for interaction, not necessarily as assistive technology, is rapidly growing into a research agenda of it's own, albeit, without notable consumer products yet in availability. The work of Nanayakkara and Shilkrot et al. spans a number of projects (not all cited here for brevity) into wearable assistive cameras to read text and recognize objects [14, 16], also occasionally serving as smartphone peripherals. Stearns et al. recently developed HandSight [19], which is geared directly at reading text with the finger. Other related work includes Rissanen et el. [15], which is a smartphone peripheral camera for natural interaction with objects, and Yang et al. that created a miniature finger worn device that reacts to surface texture [23].

Camera-augmented fingers as an approach to assistive technology were also conceptualized earlier by designers without a technical implementation. Hedberg thought of the Thimble, a device to allow reading print and also braille [7], Lee designed the Reading Finger that reads barcodes [11], and both Munscher [5] and [12] thought to use the finger as a point-and-shoot camera, literally. The enduring work of Stetten et al. on FingerSight [8], first reported in 2006, tries to create an assistive fingerworn device to help detect subtle visual discontinuities with the fingertip. Ando et al. looked to achieve the same function, however, with a fingernail mounted primitive camera [2]. Since the FingerReader's creation, other devices have surfaced that imitate the device's functionality [19].

⁷https://www.apple.com/watch/

⁸http://www.android.com/wear/

1.3 Contributions of this thesis

Together with the FingerReader hardware, this project has brought about the first mobile experience for an assistive wearable reading device of small form factor. This thesis contributes the implementation of the MobiReader experience on Android and expands upon the original experience with new types of feedback.

An evaluation of the MobiReader experience is also included as part of this thesis. Following similar prior work [19, 16], the evaluation sought to estimate the potential success of the device to aid in reading printed material. Building on the former studies, this project contributes a quantitative assessment of the complete working system and the responses from interviews with a larger pool of participants.

The remainder of this document discusses the implemented software architecture, technical and user-driven evaluations, lessons learned from the work, and future projections.

Chapter 2

Implementation

This chapter describes the implementation of the MobiReader system, placing emphasis on the implemented mobile software contribution on the Android platform. A description of the high level design is provided and then explored.

2.1 The MobiReader System

The MobiReader is a finger-worn peripheral camera device that is worn on the index finger. The following sections detail the hardware and software aspects of the system, as well as the user interaction scheme.

2.1.1 MobiReader hardware

Previous iterations of the MobiReader featured a ring in the form of a plastic shell that served to house an embedded analog camera. [16]. Bearing resemblance to the its predecessors, the MobiReader hardware has evolved to better suit a variety of finger shapes. The ring now features a smaller, 3D-printed frame and an adjustable strap, sporting an ergonomic design for adhering to the top of the finger. It also contains a considerably smaller camera module than that of the FingerReader, although not as small as the HandSight's NanEye [19]. The MobiReader, in contrast to FingerReader and HandSight, does not contain any vibration feedback capabilities and relies on



Figure 2-1: A screenshot of the MobiReader interface during usage

audio cues alone, which allows it to be smaller and monolithic.

The camera module in use is analog, therefore a USB Video Class (UVC) video encoder is included with the system. The UVC interface allows the MobiReader to connect to practically any device with USB host capabilities and a modern operating system, smartphones included. This way the MobiReader, while currently still a prototype, could be used in the future as a peripheral by anyone carrying a smart device, e.g. a phone or an Android-enabled CCTV magnifier.

2.1.2 Android application

Accompanying the MobiReader hardware, a version of the original computer vision algorithms in [16] is implemented for the Android platform. The application, pictured in Figure 2-1, serves as the main interface through which the external MobiReader hardware can be controlled as well as the source of computation for processing the incoming video frames. Through the application, multiple settings are available to the user that enables he or she to customize their reading experience. Settings can be adjusted to the user's preference, providing for the enabling and disabling of feedbacks while reading text, as well as customizing whether incoming words are read in their entirety or cut off when a new word is found. Speech rate, the speed at which words are read, can also be adjusted.

2.2 Design

2.2.1 User Experience

The ultimate goal for this project is to take the FingerReader [16] and pair it's functionality with mobile smart phones. We envision an experience in which userâĂŹs could be anywhere, in any moment, and be able to perform the FingerReader's functions.

Ultimately the end-to-end mobile experience is implemented to function as follows:

- 1. Users start by wearing the MobiReader with their index finger and installing the FingerReader pairing application on their smartphone.
- 2. To start reading text, users open the MobiReader application and plug the ring into the micro-USB slot on the smartphone. This initializes the accompanying software backend, which is immediately ready to start reading and processing incoming textual data from the ring.
- 3. Upon hovering the ring over text, the experience described in [16] will start. The phone will start providing feedback to the user about the text they are reading through the phone speakers or the plugged in headphones.

Reading Experience

In accordance to the original FingerReader implementation, a few types of feedback are used to guide the user's finger placement on a line as they scroll from left to right to read. For the mobile experience, these are detailed as follows:

- 1. Distance feedback: provides the user with an indication as to whether their finger is veering above or below the text being read. For the mobile experience, this is relayed as an audible sine wave pattern that ranges in frequency between 540Hz 740Hz. As a user's finger veers below a line, the tone approaches 540Hz and becomes louder, indicating to the user that they should steer their finger up. Similarly, the higher a user's finger veers above the line, the louder and closer to 740Hz the tone becomes, indicating the user should steer their finger down.
- 2. Angle feedback: Ideally, a user's finger would point at the text such that their finger is perpendicular to a line of text. Angular feedback informs the user that their finger is at an unfavorable angle to read a line of text. In the mobile implementation, this tone spans a frequency range of 940Hz 1140Hz. Should the user's finger turn counterclockwise to the text, a tone increasing in volume and frequency towards 1140Hz is emitted prompting the user to rotate their finger clockwise. If the finger is rotated too far clockwise, a tone increasing in volume and decreasing in frequency towards 940Hz is emitted prompting the user to rotate their finger to rotate their finger counterclockwise. When played, this feedback is overlaid on top of the distance feedback.
- 3. End of line feedback: indicates to the user that they have reached the end of a line of text. For the mobile experience this is conveyed as a repeating stock dial tone, which, to the user, sounds like a looping sequence of "beeps". End of line feedback is also used to indicate when a line is lost but is only output as a single "beep".

A number of accessibility features have been included that allow the user to customize these feedbacks. This is further discussed in section 2.2.1.

New FingerReader Feedback

During the development of this project a number of additional types of feedback were incorporated into the mobile experience as a means of improving upon the original experience.

- Improved text to speech feedback provided by built in Android libraries
- Improved distance and angular feedback now changes tone as well as volume to provide directionality

A number of changes were made in response to issues discovered in user testing. Initial tests revealed that users had issues initially identifying text, uncertain at times if they were in a state where they were tracking a line of text. To help users identify potential text the following feedbacks are introduced with the MobiReader:

- No text feedback: A constant emission of a subtle undertone indicating the user is not pointing to any text. This is presented as a square wave emitted at a low volume with a frequency of 610Hz.
- Candidate line feedback: the emission of a tone indicating the potential for lines of text when the user is scanning to identify a new line. Like the no text feedback, this feedback is presented as a square wave at a constant, low volume. It rings, however, at a higher pitch of 670Hz.

User Customization

Internal settings are available for the user to customize their mobile reading experience. Among these are:

- Toggle distance feedback
- Toggle angular feedback
- Toggle candidate line feedback

- Toggle no text feedback
- Enable queueing of detected words to prevent Text to speech from cutting words off

2.3 FingerReader on the Android Platform

The main goal for this application is accessibility. The platform upon which the computations are housed should therefore be readily available to the general user. The Android framework was selected for this project both for it's widespread popularity among users and it's expansive development libraries. The recently added support for UVC (USB video device class) devices in versions of Android after 4.3 also makes the framework compatible with hardware alterations should the mounted camera change in future iterations.

Additionally, most of the newer Android devices provide faster CPUs (and even GPUs) that provide better multicore support. The foundation for not implementing the FingerReader for mobile devices originally derived from poor performance of mobile devices at the time. Improvements in hardware have placed more current mobile devices on par with the performance level necessary to sustain the stream of data that the MobiReader provides.

2.4 Architecture

The MobiReader system consists of several components. The application provides a front end from which the user can modify application settings and preview captured camera input and interpreted words. In the background, services communicate with external camera hardware to obtain input and identify words from the input to be read out loud to the user. To manage the reading experience and provide feedback to the user, the FingerReader code sends messages to front-end modules indicating when to speak out the words and relay audio feedback to adjust the user's finger position.



Figure 2-2: High level diagram of the MobiReader application architecture. Blue components represent front-end interfaces, red represents back-end components, and green signifies messages passed in the form of Android Intents

2.4.1 High Level System Construction

Figure 2-2 depicts the high level design of the application framework.

2.4.2 Front-End Description

For a visually impaired user, the user interface need not present itself with complicated actions and small buttons that may be difficult to find or press. The MobiReader interface provides accessible controls through which the user can access the peripheral camera as well as a preferences panel through which the user can customize the types of feedback received. For the purposes of the prototype implementation, a debugging screen is presented so that a user can visualize what the external camera sees and the textual areas that the backend software analyzes.



Figure 2-3: (a) the front-end interface featuring the camera view of the MobiReader peripheral. (b) the preferences interface where a user may customize the application settings

FingerReader Interface

For the purposes of aiding the user, all meaningful actions can be controlled through directional swipes on the main screen. Swiping upwards turns on the attached MobiReader camera and begins the processing of video data and feedback. Swiping downwards stops this process and turns the camera off. Swiping right takes users to the preference menu. All other visible controls are presented for the purposes of aiding in debugging for testing purposes and present no further use to the user.

Recognizing that people with visual impairments are the primary users for this application, the camera settings can be controlled by the direction of a user's swipes on the touch screen within the mobile application. A swipe up turns the camera on and initiates the FingerReader backend to operate on incoming video data. If candidate line and no text feedbacks are enabled, their associated square waves start emitting. Swiping down turns the camera off and stops all audible feedback.

Another feature is available to the user that enables automatic connection to the camera upon its insertion to the USB port. This prevents having to perform the swiping process above to begin working with the FingerReader. Unplugging the external ring also results in the halting of the FingerReader session.

FingerReader Settings

For those aspects of customization discussed in section 2.2.1, internal settings are available for manipulation by the user through a settings menu. Through the Android preferences library, alterations to user settings are preserved between sessions, providing for user interaction with this interface only when necessary. Figure 2-3b provides a screenshot of the settings interface.

2.4.3 Backend Description

The FingerReader experience is enabled through the processing of an incoming video stream from the ring for textual data. To access raw video data from the UVC driver and operate on it efficiently, the mobile backend is written on a foundation that uses the Android native development kit (NDK). The raw video data is interpreted using a modified implementation of the original FingerReader algorithm.

Native Component

The native component of the MobiReader application reads video data frames from the peripheral hardware and operates on them to fulfill the reading experience. For debugging purposes, the native component also manipulates and updates a fragment of the UI with a view of the live camera feed and the text tracking (displayed in Figure 2-3a).

The built in support for UVC post Android version 4.3 has enabled support for reading from cameras plugged into the micro-USB slot of Android devices. As a result many open source libraries now exist that enable the easy access to reading video data from external cameras

The native component of the MobiReader application uses one such library known as libuvc¹. Through this library, video frames from the camera mounted on the MobiReader can be directly read into the native component for further processing.

FingerReader Processing

With access to the video data stream, it is a natural step to incorporate the original FingerReader algorithms to the incoming data stream. libuve provides means for overriding a callback function that is called whenever a video frame is read from the incoming stream. The same algorithms described in [16] thus run in within the native component, using the data from singular frames to track lines of text and discover potential words.

2.4.4 Computer vision algorithm

The bulk of the algorithms used for the MobiReader are the ones used in [16], however this project has made a number of additional features and improvements. In broad

¹https://int80k.com/libuvc/doc/



Figure 2-4: The state diagram from [16] for the computer vision algorithm modified to include the new/improved MobiReader states (marked in red)

strokes, the reading algorithm is a state machine that governs the process of giving feedback to the user with the following states: *No Text, Candidate Line, Reading Line, and End of Line.*

Existence of text (*No Text / Candidate Line* states) is determined by the number of qualifying character contours in the focus region, which is determined by the visible tip of the user's finger in the camera frame. If there are more than 2 qualifying characters that form a mutual baseline (tested by means of voting and fitting a line equation) the system transitions to *Candidate Line* state. In *Candidate Line* mode it will look for the first word on the candidate line via OCR.

The OCR engine, based on an Android version of Tesseract [18], compensates for the distortion caused by the angle the finger takes with the paper. If the text is at an angle with respect to the image determined by the precomputed line equation, a 2D central rotation will correct it. Thereafter, an intelligent trimming process will remove the whitespace surrounding the first word. The first word is determined by looking for large gaps in the x-axis projection of the words' image patch (reducing the rectangular patch to a single row with the MAX operator on each column), similar to [19]. For the mobile version, the trimmed patch has been reduced in sized for quick processing by Tesseract when in *Single Word* mode. OCR also does not occur on every frame, rather, only when new candidate words appear, greatly improving performance on the mobile processor.

The finger-tip detection algorithm of [16] was inefficient and expensive to execute in a mobile setting. This has been improved through the introduction of a coarse-tofine method. The algorithm now analyzes an extremely downscaled (uniform 10%) version of the normalized-RGB image to determine approximate fingertip placement. It later inspects the rough estimate in a small 100x60 pixel window to make the determination for exact fingertip location. A standard Kalman filter is introduced to counter noise in the measured fingertip point signal, which, if left unaddressed, would have detrimental effect on the stability of the algorithms down the pipeline. The result of these improvements is a much smoother finger tracking experience that removes some burdening computations from the mobile processor.

2.4.5 Thread Level Description

To optimize performance, the MobiReader application performs its operations in a multithreaded environment. The threads and their purposes are summarized as follows:

- UVC thread The backend thread is the main thread under which frames from the external camera are retrieved. Through the callback function, the computer vision algorithm operates on the retrieved frames. Should the computer vision algorithm identify a word or determine displacement from the reading line etc, feedback is relayed as an Android Intent back to the UI thread.
- UI thread The UI thread handles all the user interactions and updates to the front facing interfaces. It is also responsible for receiving messages passed in the form of intents from the UVC thread and manipulating shared variables as a result.
- Preview thread The only UI element that is not directly controlled by the UI thread is the debug preview window that renders the view from the ring. The



Figure 2-5: A depiction of the active threads running within the application. Native threads are depicted in red while threads in the Java environment are depicted in black. Black arrows signify message passing through shared memory. The green arrow signifies message passing through Android Intents. Blue dashed arrows signify the initiator of the thread.

thread maintains a queue in which frames retrieved from the UVC thread are stored and rendered to a view fragment in the front end.

- Distance Audio Thread The distance audio thread serves to output the distance audio feedback. The thread operates by continuously writing sine waves varying in frequency to an audio track that is played as output through the speakers. Based on incoming line distance measurements, the thread reacts by adjusting the frequency and amplitude of the written waves. The distance audio thread is also responsible for playing the no text and candidate line feedbacks.
- Angle Audio Thread Angular feedback is isolated in its own thread and operates similar to the distance audio thread.

To communicate between the native threads and the UI related threads, a number of messages are passed between in the form of Android Intents. The messages are used to describe events that occur within the state machine described in 2.4.4, such that the front end may provide external feedback to the user. Figure 2-5 depicts the flow of these messages and the threads responsible for delivering/receiving them.

Chapter 3

Evaluation

As part of the development of the MobiReader application, an evaluation was conducted to determine its effectiveness. A usability study was designed and conducted with ten VI persons in a lab setting. Following similar prior work [16] [19], the study sought to estimate the potential success of the device in aiding its users to read printed material. These prior investigations, however, either do not include quantitative measurements [16] or do not test an end-to-end system [19]. The following chapter discusses the user study conducted on the MobiReader implementation from Chapter 2. This work contributes a quantitative assessment of the complete working mobile system, including computer vision subsystem, as used by a larger group of visually impaired persons. Interviews with the participants provide supplemental qualitative data.

3.1 Goals of the user tests

The evaluation was conducted to assess the usability of the MobiReader system and analyze how improvements in the new system improved user experience. Statistical data would indicate potential aspects that could be further improved. The aspects to be tested were:

• User understanding: from the words read, was the participant able to extract the meaning of the piece?

- Accuracy: how many words in the piece were read in the time allotted?
- Effectiveness of mobile feedback: do the sounds and feedbacks emmitted from the mobile device fulfill their purpose in directing user's finger placement? Does it aid or annoy the user? What types of feedback most directly impact the reading experience and in what way?

3.2 Procedure

The study contained four major parts: a pre-usage questionnaire, a practice session, usage tasks, and a post-usage interview. A typical duration for a single participant was 90 minutes.

Two feedback conditions were tested within subjects: Distance (D) and Distance+Angle (D+A). In 'Distance' the user hears a continuous feedback of how far their fingertip is from the line, and in 'Distance+Angle' users also continuously hear the angle their finger makes with the line. Both feedbacks were given as sine waves of different pitches per there descriptions in Chapter 2 section 2.2.1. Each feedback condition was crossed with the tasks (5 tasks for D and 5 tasks for D+A) and fully counterbalanced to remove order bias.

To gather qualitative feedback semi-structured interviews were performed starting with a Likert scale questionnaire (14 questions), 3 open questions, and finishing with an open-ended discussion about the experience and beyond it (see Table 3.4).

3.2.1 Setup

Tests were conducted in a variety of available locations, including two conference rooms and a larger open workspace, all within the MIT Media Lab. Participants sat in an adjustable chair where they could appropriate their positioning relative to the table on which the reading tasks would occur. A lamp was mounted on the table during each test to ensure adequate lighting conditions throughout the trials.

In order to track the tester's finger placement and reaction to feedback signals each



Figure 3-1: Testing environment with P7 in midst of reading with the MobiReader

reading task is printed on a single page and placed on top of an iPad set to record the user's finger placement. This setup is inspired by a similar tactic performed in [19], however, the approach actually used a the computer vision system to drive the reading of text. This enabled tracking a user's responses to the various types of feedback through their finger location on the iPad.

An external Node.JS web application is used on the iPad to track finger position. The reading task paper is mounted and aligned on the screen using a printed anchor point so the printed text can later be aligned with the recorded touchscreen data (see Figure 3-4). Being that multiple parts of the user's hand could be touching the paper and sensed by the iPad, the application only tracks the top touch point from the user, corresponding to the touch of finger wearing the reading device. The application records the x-position, y-position, and time-stamp of each finger movement over the paper and writes that information to a text file. These points are rendered to the screen to provide additional visualization of the user's finger tracing. The reading device was attached to a Samsung Galaxy S5 Android phone placed on the table, which was emitting the audio feedbacks (see Figure 3-1).



Figure 3-2: Display of the iPad testing interface used to track the users finger position. Pages were aligned with the grey boundary and taped to tablet to secure the position. As users trace their finger over their page in the grey boundary, their finger position is recorded

3.2.2 Pre-Usage Questionnaire

To collect demographics and usage habits of the participants, a pre-usage questionnaire was fulfilled by each tester. Table 3.1 depicts the questions asked of each user.

3.2.3 Practice

The practice session included a demonstration of the device to help accustom the user to its usage. Participants were explained the meaning of all feedbacks and taught how to use them to their advantage in navigating text. Users were provided a printed text sheet with three lines of text (Figure 3-3): a single line of text at the top to teach the user how to stay on one line, and a sub sequent two lines of text further down the page

Question

What is your age?

What is the nature of your visual impairment? Is it congenital or onset at an early age? Are you right or left handed?

How do you access printed text at the moment?

What are the major hurdles in using these techniques?

What would you like to see in a new technology that would help you achieve this?

Table 3.1: Pre-usage questions.



Figure 3-3: The practice sheet used during testing

to teach the user how to find and trace a succeeding line. During the practice the participants received full support in finding the text and working with the feedbacks to stay on it.

3.2.4 Usage Tasks

During the main portion of testing, participants were given 10 reading tasks. Table 3.2 details the construction of each task. The tasks were divided into 8 news abstracts and 2 mock-business-cards. For the news abstracts, total length averaged 4.5 lines and 48.4 words (SD=4.5), and varied in font size from 9pt to 12pt. The business cards each contained 8 lines and had 22 or 23 words. Font size ranged from 6pt to 12pt in order to more accurately resemble variety in actual business cards.

Participants were given up to 5 minutes to complete a single task, and did not receive any assistance from the investigators save for encouragement and adjustments to the ring placement on their finger if it were to shift during usage. Participants

Task	Font Size	Word Count	Description
news 1	9pt	66	A news article about undersea creatures.
news 2	$10 \mathrm{pt}$	55	Article detailing new cancer treatment.
news 3	11pt	37	Article about new sensor for detecting spoiled foods
news 4	$12 \mathrm{pt}$	41	News article detailing information abaout a memorial
news 5	9pt	56	A story about a passage from India
news 6	$10 \mathrm{pt}$	46	Article detailing new materials for lighter vehicles
news 7	11pt	38	News about using machine learning for text to speech
news 8	$12 \mathrm{pt}$	48	Article about student submissions for IDEAS competition
business card 1	6 pt-13 pt	23	Mock card for a PhD candidate
business card 2	6pt-13pt	22	Mock card for a professor

Table 3.2: Usage tasks and their descriptions.

	Participant								
Task	1	2	3	4	5	6	7	8	9
1	news 4 D	news 2 D+A	news 2 D+A	business card 1 D	news 6 D+A	news 1 D	business card 1 D	news 4 D	news 7 D+A
2	news 7 D+A	news 6 D	news 4 D+A	news 1 D	news 4 D	news 4 D	news 1 D	news 1 D	news 6 D+A
3	business card 1 D	news 1 D	news 8 D	news 3 D	business card 2 D+A	news 6 D+A	news 2 D	news 5 D+A	news 5 D+A
4	news 5 D+A	news 3 D	news 3 D	news 2 D	news 1 D	news 5 D+A	news 4 D	news 6 D+A	news 3 D
5	business card 2 D+A	news 4 D+A	business card 1 D	news 8 D+A	news 8 D+A	news 3 D	news 5 D+A	news 2 D	business card 2 D+A
6	news 6 D+A	news 5 D+A	news 5 D+A	news 4 D	news 2 D	news 8 D+A	business card 2 D+A	news 3 D	news 4 D
7	news 3 D	news 8 D	news 1 D	news 5 D+A	news 5 D+A	business card 2 D+A	news 6 D+A	business card 1 D	news 1 D
8	news 2 D	business card 1 D	news 7 D+A	business card 2 D+A	news 7 D+A	news 7 D+A	news 8 D+A	news 7 D+A	news 2 D
9	news 1 D	business card 2 D+A	business card 2 D+A	news 7 D+A	business card 1 D	news 2 D	news 3 D	news 8 D+A	business card 1 D
10	news 8 D+A	news 7 D+A	news 6 D	news 6 D+A	news 3 D	business card 1 D	news 7 D+A	business card 2 D+A	news 8 D+A

Table 3.3: Task allotment for participants where D signifies a task with distance feedback and D+A signifies a test with distance and angle feedback. Participant 0 is excluded from this table.

Category	Question	Type
Perceived Efficiency	The overall experience was enjoyable	Likert
	Accessing text with the MobiReader was easy	Likert
Difficulty / Success	I was able to recover most of the words	Likert
	I was able to understand the text	Likert
	It was easy to find the beginning of the line	Likert
	It was easy to find the first line of text	Likert
Audio Feedback	I was able to follow the Distance feedback	Likert
	I was able to follow the Distance+Angle feedback	Likert
	I was able to understand the End of Line feedback	Likert
	I was able to understand the Candidate Line feedback	Likert
Comparison	Accessing text felt easier with the MobiReader	Likert
Independence	I would need the help of a technical person to be able to use the MobiReader	Likert
	I felt I needed to learn many things before being able to use the MobiReader	Likert
	I feel a greater desire to be able to read independently	Likert
Open ended	What methods did you use to work with the MobiReader?	Short answer
	What did you like about the MobiReader? What did you dislike?	Short answer
	If MobiReader worked perfectly, what would you use it for?	Short answer

Table 3.4: Post-usage questions asked after completion of all tasks.

were allowed to stop the current task at any point during the timeframe.

3.2.5 Post-Usage Questionnaire

Following the conclusion of the usage tasks, users were asked to fulfill a post-usage questionnaire about their testing experience. Questions consisted of a inquiry of about their opinions on the experience and were rated on a Likert scale of 1-5 with 1 being the highest rating and 5 being the worst rating. Users were also asked a series of short answer questions during which they provided open feedback. The questions are displayed in Table 3.4.

3.3 Participants

Nine participants (4 female, 5 male, aging 46 ± 15 years on average, range 20-63) were recruited from a ready pool of volunteers. Details of the nature of visual impairment and their text-reading habits are given in Table 3.5. An additional participant volunteered to be a pilot participant (in Table 3.5 appears as P0) and helped reduce the bugs and oddities in the system, as well as practice the study procedure. All participants received compensation for their time.

	Age	Gender	Hand	Visual Impairment	Print access
P0	61	Female	Right	Retinopathy of prematurity. Congenital. Totally blind	PC: Flatbed scanner, Kurzweil Mobile: kNFB reader, Prizmo, SayText
P1	20	Male	Right	Extreme myopia (< -12.00D)	Uses heavily corrective glasses
P2	30	Female	Right	Retinal damage. Congenital. Able to see shadows	PC: OpenBook Mobile: kNFB reader, StandScan
P3	63	Male	Right	Cancer of the retina. Totally blind.	PC: TravelScan, ScanSnap Mobile: kNFB reader, Braille.
P4	36	Female	Right	Stevens-Johnson syndrome. Onset at age 3. Some light perception.	PC: Flatbed scanner Mobile: kNFB reader, human readers.
P5	61	Male	Right	Retinal cancer. Blindness onset at age 9. Totally blind.	PC: Flatbed scanner Mobile: kNFB reader, AbiSee, Kindle, Optacon.
P6	63	Female	Right	Retinopaty of prematurity. Congenital. Totally blind.	PC: OpenBook Mobile: kNFB reader, AbiSee, Kindle, BookShare.
$\mathbf{P7}$	54	Female	Right	Macular scatoma. No central vision.	PC: ScreenReader Mobile: CCTV magnifier, magnifying glass
P8	39	Male	Right	Retinopathy of prematurity. $20/200$ right eye, left eye a prosthesis.	PC: ScanSnap Mobile: kNFB Reader
P9	34	Male	Left	Retinitis pigmentosa. Totally blind.	PC: Flatbed scanner. Mobile: kNFB Reader, human readers

Table 3.5: Details of the participants in the user study. Note: P0 was a pilot participant.

3.4 Analysis

The quantitative measurements were aligned, cleaned and analyzed as detailed in the following sections. The qualitative feedback was transcribed, coded and categorized by the investigators during the interviews and later from the video recordings.

3.4.1 Data synchronisation and alignment

Data was recorded simultaneously on both the smartphone and the iPad. On the phone a data point was recorded every time the UI thread received a message from the UVC thread encoding an instance of one of the feedback events. This included any instance in which there was no text, a candidate line was available, a word had been found, the end of line had been found, and the distance an angle feedback for when the user was tracing a line of text. When tracking a line, all feedbacks included the computed measurement for the distance and angle as well. Whenever a text to speech event occurred, the output would correspondingly log that text. All events



Figure 3-4: Aligned data from the iPad of a single session for P4, a completely blind person. The green-blue markings are a visualization of the *Distance* feedback the user received, while the red markings are when words were spoken out loud. The middle line of the text was missed, however the user was still able to recover 73% of the words in total and spent over 51% of the time line-tracking.

were marked by their own timestamp and stored in a CSV file.

The iPad recorded touches of the user's index finger to the screen during each task, tracing the location of the finger and how it moved. The differing timestamps made it necessary to synchronize the data. Using the millisecond time stamps from each event, a single time series was created by interpolating the numerical tracking data from the iPad and lining up the timestamps with the phone feedback events.

To align the 2D positioning from the iPad with the printed text, the known resolution of the iPad (132 DPI) and the on-screen offset for the printed mark in pixels was required. First offsetting then normalizing for the resolution, a script was written to programmatically align a rendered PDF of the page in 200 DPI. The results of one such alignment can be seen in Figure 3-4. For purpose of visualization alone in Figure 3-4, an offset was added to move the touch data to coincide with the text, where the original position was much lower (that accounts for the camera's center of projection and the difference between the fingertip and pad of the finger where the touch happens).

3.4.2 Extracted measurements

To assess the participants' performance, several measurements were created and compiled for each tester from their task data:

- Reading proficiency ("Consecutive Score"): This measures the amount of correctly and consecutively extracted words from a piece of text compared to the ground truth (list of known words in the text). The final score is based on a histogram on the length of correct words sequences. For example, if P1 was able to read 3 correct words in a row, the histogram value for a sequence of 3 gets increased by 1, and so on for any length of correct and consecutive words sequence. The final score for a single reading task is calculated like so: S = ∑_k H(k) * k² with H(k) being the histogram bin value for sequences of length k. This metric gives higher marks for longer sequences read, assuming long sequences create a better understanding of the text. The score for a task is normalized by the highest achievable score on its given passage: reading it all the way through, consecutively, line-by-line.
- Word extraction proficiency (*"Total Words Read"*): This counts the number of words the user was able to extract from the print in relation to the amount of words in it, without regard for order. This gives us another perspective to text comprehension from the MobiReader output, since the *Consecutive Score* measure allows for repeating sequences while this measure is robust to them.
- Feedback proficiency ("Tracking Time"): This measures the amount of time participants spent in the tracking modes discussed in section 2.4.4 (Reading Line and End of Line) in relation to the time spent in No Line or Candidate Line modes. To calculate, the amount of time a user spent in the tracking modes is extracted and divided by the cap time for performing the task.

To compute each of these measurements, a Python script was written to batch import the phone data and calculate task and cumulative scores for each participant as described above. Assuming slight misspellings do not inhibit understanding, mistaken



Figure 3-5: Individual success in reading per participant. Note: due to an error the evaluators were unable to extract timing data for P2.

words interpreted by the OCR are permitted in each of these measures by allowing for a Levenshtein distance of 2 in the match, but only for words longer than 4 characters.

3.5 Results

3.5.1 Quantitative measurements

On average the participants of this study were able to successfully extract 0.68 (SD=0.21) of the words in the text, some participants extracting an average as high as 0.81 of the text (see Figure 3-5).

The mean score of words extracted from the text proved higher with only distance feedback at 0.74 (SD=0.18) of the words being deciphered as opposed to the mean when using distance and angle feedback which was 0.63 (SD=0.22). Increasing the font size had a positive effect on the average success of extraction from the text with a mean of 0.72 for 11pt and 0.68 for 9pt. Increased font size showed a higher difference in mean when analyzing *Consecutive Score* with a mean of 0.51 for 11pt and 0.37 for 9pt font, suggesting that larger font is easier to track.



Figure 3-6: Reading success as crossed by (a) feedback type (a), and (b) font size.

The *Consecutive Score* is only a representation of the proficiency of a user in utilizing and adapting to the MobiReader. It, therefore, can only serve as a comparative measurement, rather than an absolute one. It is important to note that because of this, the *Consecutive Score* makes the variance in user capabilities negligible regarding feedback. The *Consecutive Score* using distance only feedback appeared to benefit users more with a score of 0.47 as opposed to distance and angle feedback with a score of 0.33.

The *Tracking Time* measure provided limited information about how successful users were in responding to the feedback given for tracking lines of text. The score does not necessarily assess whether the user is responding correctly to the feedback or not, it simply states how long a user is tracking a line which could have occurred if a user simply held their finger on a line without moving along the line. With this in mind, the *Tracking Time* data should not be analyzed on its own, but instead with the data gathered for each user individually in order to assess whether the user was successfully using the device.

Interestingly, the participant that spent the most time successfully tracking a line was also an Optacon user (P6). This participant spent 53% of the time in tracking mode and was completely blind as opposed to the individual with the next highest *Tracking Time* with 51% of the time spent in tracking mode who had residual sight

with severe myopia (P1).

3.5.2 Qualitative feedback

Qualitative data gathered from open ended interviews was collected through the postquestionnaire after participants concluded the tasks requiring the MobiReader. This data was divided into three parts: (1) Overall perceived success and enjoyment, (2) Perceived understanding of the audio feedback and (3) Perceived independence. While participants reported that the experience of using the MobiReader was overall an enjoyable one (2.00 SD=1.00), users reported that the MobiReader was more difficult to use compared to other reading aids (3.90 SD=1.13) and was not easy (3.45 SD=0.85). Despite this, on average the participates reported that they could understand the text (2.30 SD=0.78).

End of line feedback was the best perceived audio feedback with a mean rating of 1.50 (SD=0.92). Distance from the line was also well detected with an average rating of 2.10 (SD=0.94). Distance and angle combined with audio feedback had the lowest averaging score of 3.75 (SD=0.68). The standard deviations for these scores imply that most of the participants agreed on the level of difficulty regarding end of line and distance feedback. However, the feedback with the highest reported difficulty, distance and angle, had a lower standard deviation. This implies there was less consensus on the difficulty of perceiving the distance and angle feedback, suggesting that some users may prefer distance and angle feedback and therefore it should not be discounted as a form of feedback in the future.

The participants reported that the MobiReader did not require a high level of technical knowledge to use the device with a score of 3.55 (SD 1.31), and that it did not have a steep learning curve with a score of 3.40 (SD 1.2).

All participants except one reported that angle feedback was their least favored form of feedback and that they were confused by the combined feedback of distance and angle (N=8) (see Figure 3-6a). Most users (N=5) reported excessive arm strain while using the device and discomfort in keeping their arm and wrist straight and tense. Some users (N=3) reported the necessity to be very accurate while using the



Figure 3-7: Qualitative measurement of perceived success. 1 - best, 5 - worst.



Figure 3-8: Qualitative measurement of independence. 1 - best, 5 - worst.

MobiReader and to keep movements slight and constrained. Three users reported that they would not use the MobiReader for reading long pieces of text and the same number of participants reported that the overall reading process was slow. However, most participants (N=5) agreed that the device design was comfortable and small.

The most successful reported strategy (N=5) was to find the top line from the top of the page to work down to the next line, tracing backwards to the left to find the first word on the line. Though this strategy worked for the majority, some participants reported the inability to backtrack (N=3).

Participants also expressed distaste for the feedback methods in general; it was reported that the tone and increasing volume intended to help the user find their way back to the line of text induced more panic than suggestion. This was reflected through large movements that indicated that the participant reacted too strongly to the feedback and could not find the original line.



Figure 3-9: Qualitative measurement of understanding the audio feedback. 1 - best, 5 - worst.

3.6 Discussion

While great strides are made to create finger-wearable devices for assisting people with visual impairments in reading and accessing visual material, the current state of the field, MobiReader included, is that they still provide low accuracy and cumbersome operation preventing wide usage. The improvements in the MobiReader have increased its usability, however all the participants wished for a more accurate and smooth reading experience. P9 even went to ask us "Why are you doing this?", as he tried to explain that using the kNFB reader app is far more efficient. Faster algorithms and better user experience is still in pressing need.

It is interesting to note the distinct improvement in usage when users set out to accomplish tasks with only distance feedback enabled, rather than distance an angle. Qualitative data supplements the quantitative data in this regard as users generally favored exclusively distance feedback over distance and angle. Some users described this as being a result of difficulties in distinguishing one pitch over another. Others claimed the additional higher tones of the angular feedback made them uncomfortable, which may have led to less refined motions and inherently worse line tracking ability. The addition of a separate tone may have also added too much complexity to the experience of tracking a line, to the point that users were unable to accurately respond due to confusion and/or sensory overload. Nonetheless, these results indicate a necessity to alter the manner in which angular feedback is delivered, if at all. Something not directly tested was the lighting requirements the device imposed. Suitable testing conditions required adequate ambient lighting, so that the camera feed would provide appropriate luminance for the internal computer-vision algorithms to operate correctly. During test preparation, a flexible lamp boasting a TODO Watt bulb was placed near the participants and turned on to ensure appropriate lighting conditions for the device. At times, when the user had shifted or the reading task was not positioned properly, the light would have to be adjusted to ensure the accuracy of the device itself. In an uncontrolled environment, such actions may not be available to the user and would inhibit the reading experience. Section 3.7 goes into further details as to means of mitigating this stringent requirement.

Another aspect not tested, was the usability of the application component in conjunction with the reading system. While all camera and feedback settings were controlled by the test administrators during testing, it would be useful to identify the usability of the mobile accessibility features with visually impaired users. Section 3.7 goes into further details about potential future studies for this component.

Users expressed agreement that using the finger contains potential as a means or reading printed text, drawing connections to braille as an experiential parallel. Reflecting on the results, there is much to improve with the MobiReader. Ideas are discussed in the following section.

3.7 Future Work

The goals of this study focused on investigating the viability of each audio feedback and assessing the overall reading experience for VI people. Interesting future studies could explore user interaction with the main interface on the phone, focusing on the application usability. There is also room for investigation into the the robustness of the MobiReader when applied in more mobile contexts.

As mentioned in section 3.6, the necessity for appropriate lighting conditions bears a heavy price on the mobility of the MobiReader system. The entire reading experience hinges on the precondition that lighting is adequate in the user's environment. To mitigate this requirement, components of the MobiReader system should be updated to counter this precondition. One means of doing so would be to incorporate a front facing LED light into the hardware of the MobiReader ring to illuminate the passages the user is reading. This, however, would consume additional power. Another means would be through automatic luminance correction in the accompanying software component of the MobiReader, but this would not be guaranteed to function correctly, especially if the user attempts to read in an environment that has no ambient lighting.

Current MobiReader hardware utilizes an analog camera and therefore requires tethering to a power source to power the external UVC encoder. A future improvement that could remove this restriction and aid in the aforementioned future studies would be to use the phone's power source to power the mounted camera. This would also provide a means of mounting the aforementioned LED lamp to the front of the ring to illuminate readings.

There is the potential to remove the necessity for tethering in general by building wireless radios in to the MobiReader ring. A number of small wireless cameras exist in the public market that could be incorporated into the ring that would provide for a cordless peripheral. This bears tradeoffs in the performance and construction of the mobile component, and introduces potential setup hurdles for users. The MobiReader application would have to monitor a network stack for incoming frames which could impose latency in the rate at which the application provides feedback to the user about the text being read. The ring would also need a built in power source since it would no longer have access to the mobile smartphone battery, and would thus require charging prior to usage. Finally, the user would have to connect the ring to the phone in an initial setup phase, which may prove difficult for the primary audience of this device.

Future improvements to the application. The status of the MobiReader application provides opportunities for further improvement. A major step would be to transition the application to iOS, however, this would impose additional alterations in the MobiReader hardware to accommodate for the present drivers in iOS devices, among which UVC is not included.

Another hardware improvement, implied by one of the participants, would be to create a paired mechanical board that would shift textual pieces under the user's finger to prevent the necessity for users to track lines of text. Doing do would help to boost the durations of time during which the user is in tracking mode, but would likely not be feasible as a mobile solution.

The goals of this study focused primarily on investigating the viability of each audio feedback and assessing the overall reading experience for VI people. Future studies could explore user interaction with the main interface on the phone, focusing on the application usability. There is also room for investigation into the the robustness of the MobiReader when applied in more mobile contexts.

Taking into consideration user inhibition of the angular feedback, it may be worth considering replacing the feedback with automatic correction by the software component. Doing so would reduce the amount of effort necessary for a user to track a line, at the cost of performance on the application end. To put the usability of the MobiReader on par with other market products, this may be necessary.

It is worth looking into adjusting the tones and frequencies emitted as feedback to prevent alarming users. P7 noted recent studies involving silent, electric cars by the U.S. Department of Transportation have explored the domain of alerting sounds for the purposes of aiding visually impaired individuals in detecting approaching vehicles. For the purposes of improving the audible feedback of the MobiReader, such studies may provide promise for providing phlegmatic audio feedback.

Chapter 4

Conclusion

The contributions of this paper are:

- 1. A detailed recounting of the technical aspects of the MobiReader
- 2. Results of a user study performed with 10 visually impaired users, recording both quantitative and qualitative data
- 3. A discussion around the implications of the MobiReader as an assistive technology for accessing print

4.1 Lessons Learned

During the course of this project, the author learned much about the MobiReader and its associated fields. The main lessons are accounted for in this section.

4.1.1 Further Usage

During the later months of this project, a prototype of the MobiReader was presented during an open house at the MIT Media Lab. During this time, the project was showcased to a number of sponsors and industry delegates. Viewers had a chance to see a live demonstration of the ring and were welcomed to ask questions and try using the MobiReader for themselves. Reception of users demonstrated a generally positive response. Again, as mentioned in Chapter 3, inconsistent lighting presented a number of issues, and viewers took note of this. Picking up the device and trying it for themselves proved difficult without initial instruction as to how to use the device, a hurdle that most users wanted to go through as quickly as possible to experience the technology. Once users had adjusted the ring and identified their fingertips, most, to their content, were able to identify and scan lines of text.

Of most interest, however, were the discussions for further use that the MobiReader experience alluded to. In particular, many vendors took interest in the potential for the MobiReader to serve as a learning tool. It was suggested that, in addition to being marketed as an assistive tool, the ring could show promise in teaching the illiterate to read. Through the MobiReader direct interaction experience, a user could learn the meaning of written words and their pronunciation.

Another proposed usage was as a translation tool. Through the mobile application, users could set languages from which and to which the text should be translated. Upon reading a word of text, the application could simply perform the translation and speak the translation out loud instead. Such functionality would prove useful when traveling in foreign countries, or simply for learning a new language.

Beyond textual purposes, viewers also saw potential for the MobiReader to be used for object recognition. By altering the computer vision algorithms to perform object recognition instead of text recognition, the MobiReader could provide small children with a means of learning shapes, colors, and simple mathematics.

These ideas allude to the diverse potential for the MobiReader experience. Should such functionalities find themselves included in future iterations of the system, the user population for the device could be expanded greatly.

4.1.2 Further Testing

During the course of this project, the author obtained certification for working with human subjects through online coursework provided by the University of Miami CITI program. With this certification the author was able to participate and make personal observations during the evaluation period.

To completely assess the effectiveness of the MobiReader and other similar systems more work needs to be done. The study participant pool was small compared to the overall population with visual impairments for the evaluation discussed in Chapter 3, as needed, due to the length of tests conducted. The data collected provides thorough insight into the effectiveness of mobile feedback and reveals that improvements in this component are still a necessity.

Further data would be beneficial for assessment of the MobiReader system. Sections 3.6 and 3.7 allude to the lack of testing done to test interactions with the mobile application component. This again presents an opportunity in which more data should be collected to accurately assess the entire system interaction. Such tests could be conducted in a smaller, self-contained evaluation, one that only looks at application usability through a formal, task based, usability study. Isolating this component would allow tests to be shorter and provide opportunity to gather more testers.

4.1.3 Form Factor

Of interest in this project is the form factor under which the reading experience is delivered. Simply from personal observations within evaluations with testers, the ring form of the current peripheral bears a number of advantages and disadvantages. Working in advantage of the ring is the following:

- Size: The form factor of a ring presents itself as one of the smallest wearable accessories people wear. This provides portability and familiarity.
- Accessibility: The ring provides quick access to a mobile reading experience due to its wearable form. Rather than requiring the user to pull out an assistive device, a wearable peripheral presents an opportunity for constant, quick access. The MobiReader has the potential to perfect its experience and become so accessible due to its wearable form.

These advantages compare against the following disadvantages:

- Shifting of the ring: As noted in section 3.2.4, at times during the evaluation, the ring would shift on the user's finger, causing difficulties in reading text should the text be at too much of an angle. The positioning would have to be adjusted for the tester to put their finger in perspective, something a person with a visual impairment may have trouble doing.
- Hand positioning during usage: The lack of padding on the back portion of the ring means users must fully flex their finger and hand such that the finger is in view of the mounted camera. Users often wanted to relax their finger, causing it to curl such that the fingertip was not in view of the camera. Doing so would amount to failed attempts at reading and user would have to adjust and flex their hand to the point where it may have been uncomfortable for extended periods of usage.
- Initial positioning on finger: Users often mistakenly attempted to put the ring on backwards, not realizing which way the camera was facing. Even when correctly placed, users with shorter fingers found difficulty placing the ring far enough towards the back of their finger such that their fingertip was visible. This often resulted in further strain, attempting to straighten their hands enough to put the finger in perspective.

These disadvantages present a number of usability problems that would need to be fixed should the system have hopes of commercial acceptance. The author has considered a few solutions, both software and hardware oriented to mitigate these concerns.

From a software standpoint, it would be trivial to incorporate a new feedback type into the system. As the computer vision state machine already detects presence of fingertip within the image frame, a reasonable improvement would be to take advantage of this knowledge and emit a sound to notify the user whenever their fingertip is out of range of the camera view. This solution, while simple to include given the framework of the application implementation, bears the disadvantage of further complicates the experience provided to the user as they must now listen for an additional feedback.

Of potential interest from a hardware standpoint is the idea of altering the form factor of the ring to be embedded in a glove or partial glove. Doing so would aid in properly positioning the mounted camera on the finger and would prevent shifting were the glove to have a proper fit. The glove could also house a power source and provide haptic feedback about the correctness of the user's hand position. Moving to such a form factor would require extensive hardware redesign and thus merits further research into its advantages and disadvantages.

4.2 Android Discussion

From direct experience with the development, the following section discusses the author's perspectives of the Android implementation.

4.2.1 Conquering the native environment

Reading from an external peripheral requires direct access to file descriptors and drivers that standard Android libraries are not privy to. The decision to use the Android NDK was therefore based on necessity, as it is the only means through which Android applications can achieve desired access.

Using the native environment bears its only set of technical challenges. Native code runs in C/C++ and requires separate compilation from Android Java code.

Of particular interest from the technical standpoint was the work done to enable communications of state changes within the computer vision processing. Threads spawned within the native environment do not have access to the original Java environment, a troubling concern for the UVC thread which houses computations performed for the computer vision algorithms. This means that a standard Java listener pattern cannot be trivially implemented. An alternative means of accessing native state, would be to poll the native environment from the Java environment. This, however, would impose performance degradations (see 4.2.2), and may not lead to absolute correctness due to concurrency issues.

Fortunately, the native environment does have capabilities to create new Java environments through the overarching virtual machine for the application, enabling native code to access general Java libraries and providing an alternative means of sharing state. Through the Android Intent library, the MobiReader application sends messages about state changes in any native thread to the UI thread. The native environment can direct Android to broadcast Intents, so long as it has knowledge of the application context that the message is to be received in. At the start of the application, a reference the the main activity is passed to the native environment and stored. This reference is never altered by the Java code, allowing it to be used in the native environment for the context upon which to send Intents without the reference going stale. The MobiReader application listens for Intents broadcasted with messages detailing the state of the computer vision state machine within the native environment. Through these messages, the front end provides the correct feedback, in a timely, sequenced manner.

4.2.2 Limitations

A number of tradeoffs come into play with the transition of the FingerReader to a mobile platform. Notably performance takes a hit. While libuve provides support for reading video data at speeds up to 30 frames per second, the MobiReader software averages 7.15 frames per second. This limits the feedbacks received while scanning text and should be improved in future iterations.

The experience is not seamless for the entire system. Due to security issues, an additional confirmation step is required from the user whenever connecting to the external FingerReader camera. As mentioned in section 3.7, the presence of a cord requires the user tether to a power source, inhibiting mobility until this power source transitions to one contained within the system. The user must also find text to read in the first place. Support for fonts and languages is also significantly limited at completion of this thesis.

When evaluating testers, nearly all users possessed an iOS device. Testers claimed

the platform to be more popular than Android, a claim also supported in [13]. This presents a limitation in the audience this software could reach were it not to support multiple platforms.

4.2.3 Future Iterations

The Android software provides the means through which the MobiReader experience is delivered. This being the case it is important that the software be optimized to provide the best interactions. After evaluation, the author has put further consideration as to how the overall user experience with the application could be improved.

One potential improvement to the software that could benefit user experience would be to alter the programming to be more service oriented. For instance, instead of having an Android Activity serve as the main interface through which feedbacks are delivered both visibly and audibly to the user, usage of an Android service to perform these functionalities would enable the MobiReader experience to run in the background. This could provide for further usage of the phone's other applications, while still enabling access to the MobiReader experience.

In its current state, the application requires the user have the screen on and application running in the front end to access the reading experience. This consumes the battery and means the user cannot place the application in their pocket without fear of the screen being manipulated unintentionally. An additional improvement would be to enable this experience even while the phone is locked. Transitioning to a service based approach would provide for this approach, but the means in which the feedback is delivered may require further change, likely requiring use of Android audio libraries to stream the feedback as an audio track rather than simply playing tones.

4.3 Conclusion

The MobiReader succeeds the FingerReader presented in [16], introducing new feedbacks and user experiences, all within a new framework on the Android platform. Through evaluations and user interviews, the MobiReader has proved to extend the potential of its predecessor as a mobile assistive device, but merits further improvements and research before transitioning to a commercial product. This project contributes the MobiReader Android application, the evaluation of the system amongst users, and a recounting of lessons learned from the experience. Areas for future work include optimizing application performance, perfection of user experience, and research into untested application components. Through the MobiReader, this project contributes a new step towards improving mobile accessibility solutions for reading printed text.

Bibliography

- [1] Blindness statistics, August 2014. https://nfb.org/blindness-statistics.
- [2] Hideyuki Ando, Takeshi Miki, Masahiko Inami, and Taro Maeda. Smartfinger: Nail-mounted tactile display. In ACM SIGGRAPH 2002 Conference Abstracts and Applications, SIGGRAPH '02, pages 78–78, New York, NY, USA, 2002. ACM.
- [3] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. VizWiz: Nearly Real-time Answers to Visual Questions. In Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology, UIST '10, pages 333–342, New York, NY, USA, 2010. ACM.
- [4] E. E. Fournier d'Albe. On a type-reading optophone. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 90(619):pp. 373–375, 1914.
- [5] David MAijnscher. Every Stalker's Dream: Camera ring, October 2007.
- [6] Yasmine N. El-Glaly, Francis Quek, Tonya L. Smith-Jackson, and Gurjot Dhillon. It is Not a Talking Book;: It is More Like Really Reading a Book! In Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '12, pages 277–278, New York, NY, USA, 2012. ACM.
- [7] Erik Hedberg and Zack Bennett. Thimble There's a Thing for That, December 2010.
- [8] S. Horvath, J. Galeotti, Bing Wu, R. Klatzky, M. Siegel, and G. Stetten. FingerSight: Fingertip Haptic Sensing of the Visual Environment. *Translational Engineering in Health and Medicine*, *IEEE Journal of*, 2:1–9, December 2014.
- [9] Shaun K. Kane, Brian Frey, and Jacob O. Wobbrock. Access Lens: A Gesturebased Screen Reader for Real-world Documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 347–350, New York, NY, USA, 2013. ACM.
- [10] Shaun K. Kane, Chandrika Jayant, Jacob O. Wobbrock, and Richard E. Ladner. Freedom to Roam: A Study of Mobile Device Adoption and Accessibility

for People with Visual and Motor Disabilities. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '09, pages 115–122, New York, NY, USA, 2009. ACM.

- [11] Hansub Lee. Finger Reader, September 2011.
- [12] Andrew Liszewski. Ubi-Camera, March 2012.
- [13] J. Morris and J. Mueller. Blind and deaf consumer preferences for android and ios smartphones. In P. M. Langdon, J. Lazar, A. Heylighen, and H. Dong, editors, *Inclusive Designing*, pages 69–79. Springer International Publishing, 2014.
- [14] Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. EyeRing: A Finger-worn Input Device for Seamless Interactions with Our Surroundings. In *Proceedings of the 4th Augmented Human International Conference*, AH '13, pages 13–20, New York, NY, USA, 2013. ACM.
- [15] Mikko J. Rissanen, Owen Noel Newton Fernando, Horathalge Iroshan, Samantha Vu, Natalie Pang, and Schubert Foo. Ubiquitous Shortcuts: Mnemonics by Just Taking Photos. In CHI '13 Extended Abstracts on Human Factors in Computing Systems, CHI EA '13, pages 1641–1646, New York, NY, USA, 2013. ACM.
- [16] Roy Shilkrot, Jochen Huber, Wong Meng Ee, Pattie Maes, and Suranga Chandima Nanayakkara. FingerReader: A Wearable Device to Explore Printed Text on the Go. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2363–2372, New York, NY, USA, 2015. ACM.
- [17] Roy Shilkrot, Jochen Huber, Jurgen Steimle, Suranga Nanayakkara, and Pattie Maes. Digital digits: A comprehensive survey of finger augmentation devices, 2015.
- [18] Ray Smith. An overview of the Tesseract OCR engine. In *ICDAR*, pages 629–633, 2007.
- [19] Lee Stearns, Ruofei Du, Uran Oh, Yumeng Wang, Leah Findlater, Rama Chellappa, and Jon E Froehlich. The Design and Preliminary Evaluation of a Finger-Mounted Camera and Feedback System to Enable Reading of Printed Text for the Blind. September 2014.
- [20] Deborah Stein. The optacon: Past, present, and future, September 1998.
- [21] Michael Tomasello, Malinda Carpenter, and Ulf Liszkowski. A new look at infant pointing. *Child Development*, 78(3):705–722, 2007.
- [22] Peter White. books Digital may not be for everyone. but blind people, they're revolution, August 2012.for true \mathbf{a} http://www.theguardian.com/commentisfree/2012/aug/17/peter-white-booksbraille-ereaders.

- [23] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Magic Finger: Always-available Input Through Finger Instrumentation. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, pages 147–156, New York, NY, USA, 2012. ACM.
- [24] Debra Yarrington and Kathleen McCoy. Creating an Automatic Question Answering Text Skimming System for Non-visual Readers. In Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '08, pages 279–280, New York, NY, USA, 2008. ACM.
- [25] Hanlu Ye, Meethu Malu, Uran Oh, and Leah Findlater. 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, pages 3123–3132, New York, NY, USA, 2014. ACM.
- [26] Chucai Yi and Yingli Tian. Assistive text reading from complex background for blind persons. In *Proceedings of the 4th International Conference on Camera-Based Document Analysis and Recognition*, CBDAR'11, pages 15–28, Berlin, Heidelberg, 2012. Springer-Verlag.