SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING

Date: June 6, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Yang Li
Luis Abraham Millan
David Blake Tsuzaki

ENTITLED

Portable Reading Assistant Headset for the Visually Impaired (PRAHVI)

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING

_____
Thesis Advisor

_____
Thesis Advisor

_____
Department Chair

_____
Department Chair

Portable Reading Assistant Headset for the Visually Impaired (PRAHVI)

by

Yang Li
Luis Abraham Millan
David Blake Tsuzaki

Submitted in partial fulfillment of the requirements
for the degrees of
Bachelor of Science in Computer Science and Engineering
Bachelor of Science in Electrical Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 6, 2017

Portable Reading Assistant Headset for the Visually Impaired (PRAHVI)

Yang Li
Luis Abraham Millan
David Blake Tsuzaki

Department of Computer Engineering
Department of Electrical Engineering
Santa Clara University
June 6, 2017

ABSTRACT

Most products in the domain of improving the visually impaired textual understanding focus on the direct translation or dictation of text that is in front of a user. Seldom focus on any type of textual understanding that goes beyond literal translation. In this report, we are documenting the implementation of a novel wearable device that allows the visually impaired to have a better understanding of the textual world around them by having a system learn a textual understanding for them and then providing more significant and natural feedback based on a users semantic queries regarding the text at their gaze.This document includes the requirements, design, use cases, risk tables, workflow and our projected development timeline for this device we are developing. This report also provides a way for us to review our progress and justify decisions we make as we advance in the development phase.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Communication is the hallmark of our interactions as humans, occurring across different media, platforms, and entire paradigms. Much of the information we consume on a daily basis is provided in very specialized media, such as a newspaper or a billboard, that cater to only one specific sense, such as vision. This presents a particular challenge for individuals with sensory disabilities. Every day people absorb visual, sonic, and touch information from their surroundings. The visually impaired rely on a heightened sense of sound and touch to obtain information and are hindered from being able to easily obtain data from visual texts, such as posters, newspaper, fliers, etc. This hindrance not only affects the ability to obtain important textual information, such as warning or caution signs, but it also statistically raises the likelihood of unemployment.

1.2 Current Solutions

The Braille alphabet has been one of the most common aids in bridging the gap between textual information and people with visual disabilities. However, one of the problems with Braille is that it depends on text being translated and printed on a medium that the user can touch. More importantly, Braille suffers from a low literacy rate within the visually impaired community because it requires teachers with specialized training, a luxury that is not always available at public schools in the U.S.

There are many products on the market that serve as an alternative to Braille. One example is the FingerReader by the MIT Media Lab, an electronic device that dictates the text the user touches in a document. However it can only read font that is of 12-point that the user can physically touch.

Another example, the OrCam MyEye, is a dedicated headset that also performs live text dictation. However, it is priced at $2,500 and still requires a gesture input that assumes the user has some visual capability. There is a general issue with both products directly feeding the user everything from the text content. People who are not visually impaired can skim the content or skip around to get a brief idea of the content; the visually impaired must finish listening to the entire passage to understand the whole content.

1.3 New Solution

To address this problem, we are proposing to design an assistive Optical Character Recognition (OCR) system consisting of a portable headset that captures a feed of the users surroundings, a front-end mobile application that performs live OCR of the text within this feed, and a back-end framework for building a model of textual understanding. This system is capable of reading aloud the key points of the text the user is positioned to gaze at. By using haptics and computer vision, we address the shortcomings of current solutions by allowing the user to focus on text both as near as a handheld newspaper and as far as a billboard. By using a mobile phone for processing, rather than dedicated hardware, we address another key shortcoming of the current solutions by keeping the cost of the device down and allowing the system to build a model for better dictation in the future. With this system, we hope to address one of the biggest daily challenges of individuals who are visually impaired, a very underserved segment of our society.

Chapter 2

Requirements

The Requirements section presents a categorized and itemized list of project requirements. Categories include Functional and Non-functional Requirements and Design Constraints. Functional requirements define what must be done, while non-functional requirements define the manner in which the functional requirements need to be achieved. Both categories have sub-categories, determined by the importance of a given requirement. Design Constraints are similar to non-functional requirements but constrain the solution and not the problem.

2.1 Functional Requirements:

- Critical
  - have sensors to communicate with the user and detect the users surroundings
  - communicate with the user through haptic feedback and voice dictation
- Recommended
  - have a learning model to tag specific instances of text and symbols and improve overall recognition

2.2 Non-functional Requirements:

- Critical
  - easy and intuitive to recognize text in the users environment and dictate it to the user
  - light and untethered from a large computing system
  - conform to the Federal Communications Commission guidelines on wearable devices[1]
- Recommended
  - maintainable for future usage and/or upgrades
- Suggested
  - generally aesthetically pleasing so that it does not draw too much attention from the users surroundings

2.3 Design Constraints

- The main device is a wearable headset whose hardware is self-contained
- The main devices main communication with the outside world is through a smartphone
- The main device communicates primarily through sound and touch, and not through vision

[1] https://www.fcc.gov/general/ingestibles-wearables-and-embeddables

Chapter 3

Use Cases

The Use Cases section defines specifics regarding the main, expected interactions between users and the system.

Figure 3.1 shows the list of use cases that we would like the final product to have, due to time constraints we only implemented the "Reading" use case.

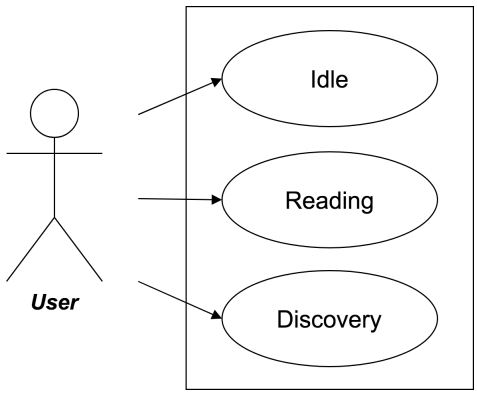

Figure 3.1: Use Cases

3.1 Idle

Goal Put the device to sleep to preserve power.

Actor User

Preconditions Device must be turned on

Steps User set the device to idle

Postconditions Disconnect connection to headset and server

Exceptions N/A

3.2 Reading

Goal Identify and obtain large amount of text and provide feedback to user

Actor User

Preconditions Device must be turned on and set to Reading Mode

Steps User stare at the document he will to know and order the device to capture the image

Postconditions Audio feedback send to user

Exceptions Unstable input, such that the motion of the headset is outside the threshold

3.3 Discovery

Goal Inform the user about potential point of interest around him or her

Actor User

Preconditions Device must be turned on and set to Discovery Mode

Steps User need to move his or her head around to capture the surrounding

Postconditions Audio feedback send to user

Exceptions Unstable input, such that the motion of the headset is outside the threshold

Chapter 4

Activity Diagram

Figure 4.1 is the activity diagram that shows the flow of actions of the user. During each decision stage, the system will prompt the user for actions.

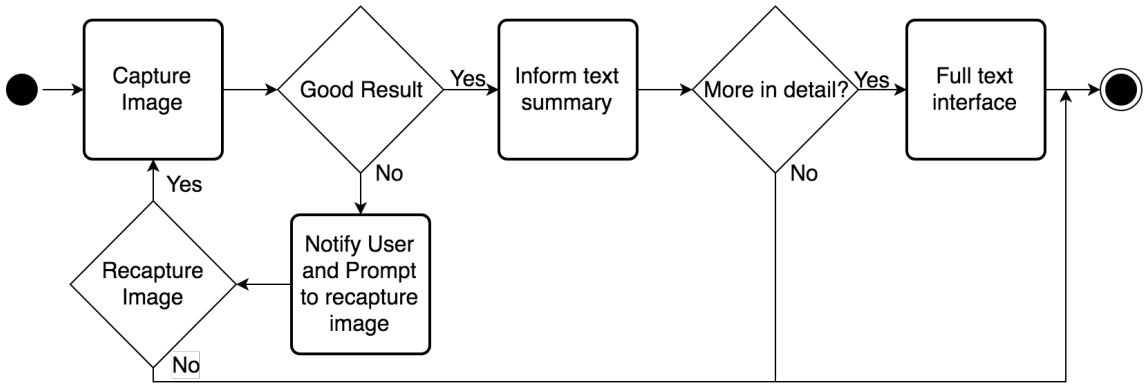

Figure 4.1: Activity Diagram

Chapter 5

Technologies Used

5.1 Tesseract Optical Character Recognition Engine:

An open source OCR engine licensed under Apache License, Version 2.0[1]. It is one of the most accurate open source OCR engines currently available.

5.2 OpenCV:

An open source library of programing functions mainly aimed at real-time computer vision. Licensed under BSD license[2].

5.3 Flask:

Flask is a lightweight python web framework which PRAHVI's backend is built on top of.

5.4 Nginx:

Nginx is a the proxy server technology that is used to deploy our Flask web server and expose it via the internet.

[1] https://www.apache.org/licenses/LICENSE-2.0.txt
[2] https://en.wikipedia.org/wiki/BSD_licenses

Chapter 6

Architecture

6.1 Architectural Diagram

This system has a data-flow architecture (see Figure 6.1), starts from "Headset" and move clock-wise to "Audio". The reason of choosing this architecture is because there are constant inputs received by the system, and the inputs in general goes through the same route within the system. The data-flow architecture also has build in concurrency, which can speeds up the process, especially the platform of the product is embedded.

6.2 Hardware Interface

PRAHVI, itself, is a wearable headset device that attaches to a typical pair of glasses or can be manufactured as a single assembly. PRAHVI is comprised of a Raspberry Pi Zero compute board coupled with a standard Raspberry Pi Camera. The modular nature of these parts allows for future

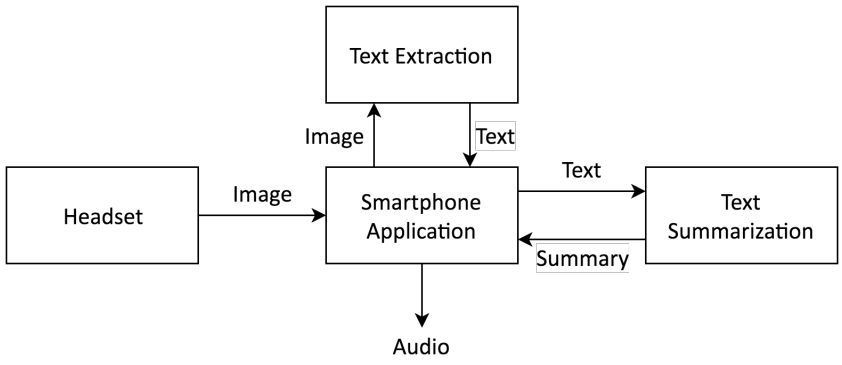

Figure 6.1: Architectural Diagram

expansion and easy repairs for the user and for other developers. The Raspberry Pi Zero, in particular, was chosen for its low power consumption and wide application flexibility. Once the device is connected to the user's smartphone and the accompanying app is opened, PRAHVI powers up and immediately begins communicating with the smartphone.

6.3 User Interface

The user interface of PRAHVI is mainly comprised of a gesture pad that allows for text translation and navigation with minimal use of visual and audible cues. Once the user connects the headset and opens the app, PRAHVI immediately begins scanning the environment, adjusting the camera parameters to find a set point that produces a clear, usable image. Swiping the gesture area allows the user to navigate the text in realtime, with the smartphone providing haptic feedback. Tapping the gesture area starts the translation process, as PRAHVI captures an image and sends the image to the backend, before signaling to the user a quick summarization of the text and then a full translation.

6.4 Backend

The backend of PRAHVI is a flask web application. It holds all the functionality related to the computer vision, optical character recognition, and text summarization related tasks PRAHVI requires.

The interface of the backend is a simple set of HTTP endpoints which are summarized as follows:

The PRAHVI iOS application makes calls to these endpoints in order to translate an image to text as well as summarize the text using the TFIDF.

Due to the limited compute resources in our mobile device, having the backend end hosted on a server allows for a faster processing time as noted in our testbench.

6.5 System Flow

Once the camera captured the image and send the image to the smart phone, the image will pass through the pre-processing stage to detect whether the image will be processed or not. Once the image passed the pre-processing stage and is also processed for text, it is then output as audio feedback.

6.6 Text Extraction

6.6.1 Image Pre-processing

When the smart phone application received the image, the image is then passed through 2 filters. The first filter detects the blurriness of the image. If the image is blurred, the phone will reject the image because it is hard to extract useful information from a blurred imaged. If the image passed the blurriness test, it will then be used to compare to the previous detected image for similarity. If the image is similar (or the same), the image will also be rejected, this is because the information of the same document is stored in the device from the previous capture. These 2 filters help to improve the efficiency of the system and save time waited by the user.

Blurriness Test

To test for blurriness, variance of Laplacian is used. The image is treated as a 2-dimensional matrix, and the variance of Laplacian of the matrix is generated. The variance of Laplacian of the image is then compared to a threshold value (the threshold value used for this system is 50). If the variance of Laplacian of the image is lower than the threshold, then the image is blurred, otherwise, it is not.

Similarity Test

The similarity test uses the accelerated-KAZE (AKAZE) local features matching[1]. The AKAZE local features matching returns a list of matches between the two images. If the number of matches is above the threshold (the threshold value used for this system is 1000), then the images is said to be similar.

6.6.2 Image Processing

When the image passed the blurriness test and the similarity test, the system will try to detect the text area in the image and extract the text area. The extracted text area is then send to the Tesseract Optical Character Recognition (OCR) Engine to convert the image to computer encoded characters.

Text Area Extraction

To extract the text area, a copy of the image is first blurred (Gaussian Blur is used in implementation) to detect the edges in the image (Canny Edge Detection is used in implementation). From the edges of the image, contours are then collected and sorted in descending order based on their area.

[1] http://docs.opencv.org/3.0-beta/doc/tutorials/features2d/akaze_matching/akaze_matching.html

From the list of sorted contours, the biggest contour that can be approximates to a quadrilateral is identified as the text area. The text area is then applied with a perspective transformation to convert the text area to a rectangle such that as if the document is scanned. This increases the accuracy of the result from Tesseract OCR Engine.

Text Detection

The transformed text area is processed by Tesseract OCR Engine to identify the bounding boxes of the text in the image and convert the image to computer encoded characters. The computer encoded characters is then processed to extract key featured and feedback to the user.

6.7 Text Summarization

In order to provide users with a summary of the text we have extracted we use an algorithm called Term Frequency-Inverse Document Frequency (TFIDF) .

The goals of TFIDF are to obtain statistically important keywords from a text article. It does this by giving each word in the target document a score based of two statistics, the word frequency and the inverse document frequency. The word frequency is simply just the number of times the word appears in the target document and the inverse document frequency is calculated by taking the log of the total number of documents in the corpus divided by the number of documents that the word appears in.

The final score for each word is calculated as follows: tf * idf

The rationale behind the tfidf algorithm is to reduce the importance of words that appear often yet have no overall significance. Examples of such words are: the, and, is, etc.

The reason why tfidf was chosen for our text summary is because it's one of the more popular and well known term-weighting schemes, it's easy to implement, and once it's set up it is computationally fast.

The corpus of documents are gathered by us. Our goal was to accumulate documents in the news domain so that way our text summarization would be inline with the domain of PRAHVI functional requirements. Our strategy was to build our corpus by scraping the top online news repositories for all of there news articles.

We took the 50 top online news websites according to the following internet article and we used a python library called newspaper to gather the body text every article currently live on the site.

Once our corpus was collected we precomputed the inverse document frequencies for each term in our corpus.

Chapter 7

Design Rationale

7.1 Architecture

- Hardware
  - Wired video module to the phone
    - Low-Latency, and therefore improving the systems responsiveness when providing feedback to user
  - Choice of Haptics + Sonics Feedback
    - Haptics allow us to better communicate spatial information regarding their gaze and important text or a users text of interest
    - Sonics are natural way of communicating textual information to a user
- Software
  - Image processing before Optical Character Recognition
    - Current Tesseract OCR Engine does poorly with text that is skewed, and therefore the image processing will allow us to deskew the image as best as possible to improve performance of the OCR engine.

7.2 Technologies

- Tesseract Optical Character Recognition Engine
  - Currently, the most accurate open source solution to optical character recognition.
  - Highly documented with academic papers written about its architecture
  - Constantly being developed by a community of developers, so if we run into problems we have a community to ask questions to
  - Has both a C (python wrapper) and a C++ API
  - Supported by Google
- OpenCV
  - De Facto standard software libraries for computer vision
  - Lots of developer support
  - Open source
  - All of its data structures are compatible with Tesseracts API

Chapter 8

Testing

The following describes how the product is tested.

8.1 Alpha Testing

8.1.1 Function Testing

During function testing, each part of the system is tested individual during development time. The following are some examples of function testing performed:

- Graphical input from camera
- Corresponding OCR input and output
- Summary feedback to user

8.1.2 System Testing

The system is tested as a whole. The test is focus on where all modules and functions within the system cooperating with each other, and whether the system functions as a whole.

Performance

The performance of the system is tested with a test set of 30 images (see Figure 8.1) with both versions of Tesseract OCR Tesseract 3.05.00[1] and Tesseract 4.00.00[2].

[1]https://github.com/tesseract-ocr/tesseract/wiki
[2]https://github.com/tesseract-ocr/tesseract/wiki/4.0-with-LSTM

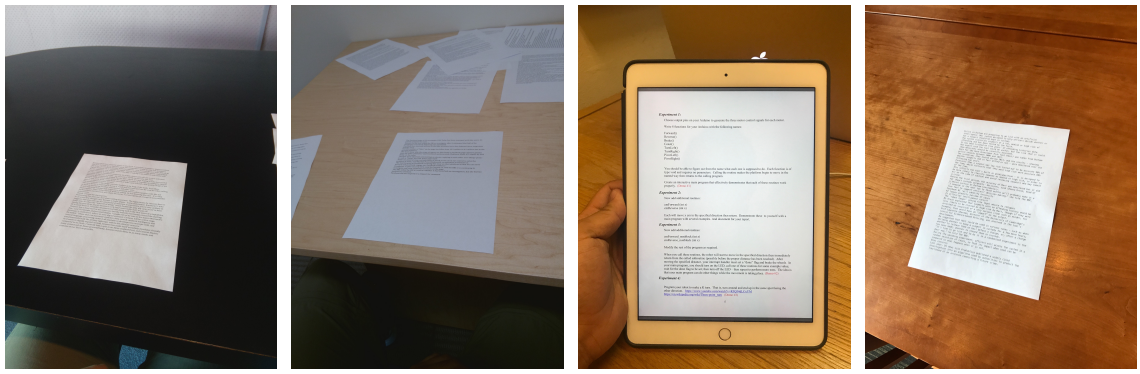

Figure 8.1: Some testing images used

Table 8.1: Performance Analysis

	Accuracy (%)		**Response Time (Sec)**	
	Mean	Standard Deviation	Mean	Standard Deviation
Tesseract 3	35.93	21.37	11.05	4.68
Tesseract 4	91.07	7.24	3.07	0.60

The document in the test images are collected from BBC News (www.bbc.com), this way the ground truth of the document (the original computer encoded characters) can be used to compare with the result of the system.

The test images cover a variety of different backgrounds, motions, blurriness and brightness. The test results (see Table 8.1) show that Tesseract 4 provides significant better results than Tesseract 3, even though Tesseract 4 is still in alpha stage.

8.2 Beta Testing

The product will be tested by visually impaired students on campus of Santa Clara University for actual user feedbacks.

Chapter 9

Risk Analysis

The Risk Analysis table defines a potential set of risks that our group could face, as setbacks to timely progression towards our finished project. For each risk, there are two potential consequences, a probability value (0→1), a severity value (0→10), an impact value (impact=probability*severity), and two potential mitigation strategies. The risks are ordered from greatest to least impact value (top-to-bottom).

Table 9.1: Risk Table

Risk	**Consequences**	**Probability**	**Severity**	**Impact**	**Mitigation Strategies**
Illness	-Work distribution becomes uneven -Project progress delayed	0.5	4	2	-Get proper rest -Stay hydrated
Loss of code	-Significant additional work -Notable setback in project progress	0.2	8	1.6	-Careful organization -Use version control system
Planned implementation failed to work	-Mid-cycle design decisions/changes required -Additional coding required	0.2	7	1.4	-Careful planning -Good communication
Schedules do not mesh	-Pace of work is potentially slower -Communication (of changes/plans) is not as fluid/immediate	0.7	2	1.4	-Careful planning -Earnest communication
Designed product does not meet expected product	-Deduction in grade -Inter-team disappointment	0.1	7	0.7	-Careful planning -Frequent project check-up meeting

Chapter 10

Ethical Analysis

The primary goal of PRAHVI is to help individuals with visual disabilities more easily integrate into society. With millions of Americans living with some level of blindness, there was an opportunity to significantly affect a broad set of individuals with some very specific challenges. Working with the university's disabilities office, we discovered typical use cases involving navigating one's indoor environment, identifying and reading texts that have no digital equivalent, and independently interacting with one's outdoor environment. These simple tasks help individuals become more productive, opening broad opportunities for work and fuller lives. Improving any single one of these daily tasks vastly improves the quality of life for individuals in this segment.

Through our ethical analysis we discovered that one of the most important facets successful engineers have is the ability and willingness to observe their work in the broad scope of the people they affect. Although the requirements, design, and implementation of PRAHVI changed during the course of its development, the team was constantly guided by the factors that would deliver the best experience to our target users given the time and resources we have. Most notably, this included prioritizing features with a sense of those that could be completed in a reasonable timeframe while driving our primary objective. This mindset ensures the success of the project in the face of unexpected conditions by focusing on the desires of the stakeholders involved.

Although PRAHVI is classified as an assistive device, its usage comes with ethical implications of its operation that we took into account during the development process. The main ethical implications are ultimately predicated on the newfound independence users gain by using PRAHVI, at the potential expense of privacy or inaccurate translation. In terms of privacy, a simple case could arise when PRAHVI

is used to translate sensitive information such as a bank statement or an address. As PRAHVI dictates the text it reads, the translation could be accidentally read to bystanders as well as the user. A more complex case could involve a malicious actor gaining unauthorized access to the device, by way of "hacking" it, and in turn record the sight and whereabouts of the device during usage. Such information could compromise the user's sensitive information or place them at risk of other attacks such as fraud or assault. These factors were taken into account during the development of PRAHVI and its software architecture to ensure that the risk for stakeholders is minimized.

Chapter 11

Sustainability

Sustainability is an essential factor in delivering a product that will benefit peoples lives and create value. During the design of PRAHVI, we have identified multiple points of sustainability through the lifecycle of the product: from its environmental costs, to its user interaction, and its economic viability. By evaluating these facets, we can draw a concrete triple bottom line to evaluate the effectiveness of the product in the context of an actual business model.

As a product, PRAHVI is designed primarily using off-the-shelf components that have been developed at scale both to reduce costs and to minimize its environmental footprint. Its construction consists mainly of a PCB board, a camera module, a plastic enclosure, and a cable to connect the product to the users smartphone. The board used in the current version is known as a Raspberry Pi Zero, a lightweight, low-power board designed for mobile applications. This means that PRAHVI operates using less than 100mW and can be powered entirely by a smartphones accessory port. The electrical components are manufactured in compliance with the global Restriction of the Use of certain Hazardous Substances (RoHS) regulation. Additionally, each company we have sourced components from have publicly pledged their products to be free of conflict materials and use minimal amounts of rare-earth metals. This allows us to minimize the environmental impact of both construction and disposal of the product by accounting for its individual materials and following federal and international procedures. The enclosure is constructed using a 3D printer with ABS plastic material. Although this material is not biodegradable, using a pure ABS filament and designing the product to be modular allows a recycling entity to separate the components and recycle the ABS plastic. Overall, we anticipate a single PRAHVI unit to last the life cycle of the users smartphone, typically 2-3 years. This accounts for future feature

upgrades and the general durability of the product against normal wear and tear. By accounting for each of these factors during the design phase of our project, we can minimize PRAHVIs overall environmental footprint.

In terms of social sustainability, we have chosen a very clear demographic that has been largely untapped for innovation making PRAHVI a promising product for this field. PRAHVI is designed to assist users with visual disabilities navigate their visually-oriented environments, from casual reading to discovering signs and posters. The use cases it presents are context-specific but very familiar to those who struggle with these tasks on a daily basis. Because we are tailoring the interface of the product to individuals with partial or full visual disability, making the product intuitive presents a unique challenge for us. By crafting an interface that communicates primarily through sound and haptic feedback, we believe that the product will be intuitive and useful for the user. Furthermore, we hope to begin testing the product through usage exercises and potentially real-world user tests. However, creating a product for this niche also draws the factor that users will create an implicit dependence on the product. Should the user begin entering high-risk environments on their own, such as navigating a busy street, the stakes for failure could mean the difference between life or death. Therefore, for the first few iterations of the product, we would advise users only to use the product in a controlled environment with minimal hazards and many safeguards. Overall, we hope that PRAHVI can add significant value to a users daily life with the objectives we have set, using the technologies and sense of interface we have developed.

The final metric for a products viability involves the economic sustainability, particularly in a world saturated with electronic assistance devices. By utilizing off-the-shelf components and relying on a smartphone that users in this demographic typically already have, we have made strides in minimizing the cost of the product to well below current solutions. Our target cost of the product is to be less than $200, which is a fraction of the closest competing product, OrCam, which is priced at $2500. Much of the cost for such devices is in the software and the processing unit, as these devices are typically designed to be standalone. During the design phase of our project, we studied the demographic of individuals with visual disabilities and found that many typically own and use a smartphone on a daily basis. This means that we can safely trade off a small measure of convenience with a standalone device for the cost savings of using a processing unit users already own. Additionally, this drives down the cost and frequency of future upgrades, as the devices processing power is upgraded for free with every

new smartphone a user purchases. This means less revenue is spent on developing and manufacturing new PRAHVI processing modules. As a product, most of the revenue would go to the material cost of the components and development and testing of the software. Additionally, the retail cost to the end user can typically be augmented by support from their insurance providers. In the future, PRAHVI may be manufactured entirely using a custom PCB board and custom hardware that, manufactured at scale, would further drive down costs while delivering a more integrated product. From an economic standpoint, PRAHVI is an effective product in this segment, especially compared to competing products, by using a careful mix of tradeoffs that overall benefit users.

As a product, we hope that PRAHVI meets or exceeds the triple bottom line to remain a fully-sustainable product. By identifying a key niche ripe for innovation, then sourcing parts and development in an environmentally and economically responsible way, we envision a life cycle that helps the product remain viable for many iterations. With each iteration, we also hope that the product can execute fixes and improvements that make it more useful for users and even expand its target audience. These goals overall help create the framework for a transition from a simple design project into an actual product.

In a world with increasingly limited natural resources and a larger focus on industrial impact on our environment, sustainability is a significant part of research and development. During the design of PRAHVI, we have identified processes, components, and the sourcing of these components to evaluate its environmental impact. As a product that spans the resources of multiple industries, we recognize that our products lifecycle includes many stakeholders and resources, from manufacture, to daily usage, and final disposal.

The construction of PRAHVI begins with its components, their sourcing, and overall assembly. The primary component of the device is its printed circuit board (PCB). For research purposes, we have used an off-the-shelf board known as the Raspberry Pi Zero. This board consists of plastic for the board itself, laser-etched copper traces, and components with varying amounts of copper, silicon, and gold. In compliance with the global Restriction of the Use of Certain Hazardous Substances (RoHS) regulation, the Raspberry Pi is manufactured without the use of conflict materials and minimal use of rare-earth elements. In addition, the camera module manufactured by Sony Inc. is manufactured under stricter regulations that replace many materials, such as those that go into the imaging sensor, with more environmentally-friendly, albeit slightly more expensive alternatives. We found that the small

form factor of the Raspberry Pi not only makes the product more portable, but holistically uses fewer materials to manufacture, further driven down by the large scale of the Raspberry Pi, while still meeting our quality and performance requirements. Although other boards and modules were evaluated, most are manufactured under small-scale operations that use more resources or did not meet our requirements. The case of the product is manufactured using a 3D printer with ABS plastic material. This material is not biodegradable, and must be recycled at the end of the products lifecycle. During the design phase of this project, we selected what we believe are the optimal components and materials for PRAHVI from a performance and environmental standpoint.

As a holistic product, PRAHVI introduces challenges to managing energy consumption from manufacture, to delivery, and daily use. The parts used in PRAHVI are sourced primarily from China. With careful design and planning, we consolidated our parts orders into three stages and from a single supplier to ensure that we minimized the impact of transportation in the product. The case, which is manufactured using a MakerBot Replicator 2X, is the only part we manufacture ourselves. For research purposes, using a 3D printer significantly reduces the energy and resource overhead of a professional manufacturer, while providing a representative component of the final product. During use, we anticipate that PRAHVI will be powered entirely using a smartphone device, removing the need for a separate battery. Its small form-factor and ARM processor allows PRAHVI to operate with minimal power use. We further these energy savings by defining clear contexts in which PRAHVI is in a passive sleep mode and when it is in an active scanning mode. These practices combined help to minimize the overall energy footprint of PRAHVI as a product.

Finally, although PRAHVI is designed with longevity of the product in-mind, we incorporated its transition out into the design process. PRAHVI is made with standard components, each of which can be easily replaced. We anticipate that PRAHVI can be used throughout the standard lifecycle of a smartphone, around two years. This accounts for component failures, the likelihood of damage resulting in a system failure, and required feature upgrades for new versions of the smartphones software. At the end of the devices lifecycle, the components of the Raspberry Pi Zero can be easily extracted and recycled through standard protocols. In addition, the case is entirely constructed from ABS plastic that can also be recycled and repurposed. These considerations help ensure that PRAHVI is built to last with the users needs as well as transition out of use in a sustainable manner.

Chapter 12

Conclusion

In this project we designed and implemented a novel and cost efficient device that assists the visually impaired.

Our device allows users to navigate text by taking a picture of an article of text that they are gazing at, translate this image to text, and finally provide a summary of the text to the user.

Costs were kept low by working with general purpose computing hardware such as the Raspberry Pi Zero, and the ubiquitous smart mobile devices.

We primarily focused on specific domain of news articles that the software works well in. In addition, we focused on lighting scenes with have sufficient lighting , and we targeted a font range of 10 to 100 pixels.

In the future, we would like to make our system extensible to more text domains and perform more robustly in harder lightling situations.

Chapter 13

Appendix

13.1 Sample Testing Documents

13.1.1 Can Amazon's assistant stay on top?

http://www.bbc.com/news/technology-39853718

Original Text

Amazon surprised everyone when, in late 2014, it unveiled a standalone digital assistant that was not only good, but blew away the competition in both quality and aesthetics. The Echo - a cylindrical speaker with microphone - now accounts for just over 70% of all digital assistant use in the US, leaving its nearest competitor, Google Home, well behind. It's an important new market, even if the idea of talking to an object in your home still comes uneasily to many of us. In a new report, Emarketer estimated 36 million Americans will use a voice-activated assistant at last once a month - an increase of 129% on this time last year. Amazon, as I mentioned, already has the lion's share. It's now hoping to echo (sorry) that success with its latest effort which we could see as early as Tuesday, according to reports. AFTV.com, a site with a solid track record of leaks, said it found a low-quality image of the device on Amazon's own servers. The authenticity of the image was later backed up by king-of-the-leaks, Evan Blass. The new device is expected to house a 7-inch touchscreen and can be used for video calling, as well as displaying weather information and other data. It will help plug that gap that many voice assistant users will be familiar with, like not knowing how long a timer has left without asking. Or just knowing the time - it's a step backwards to not just look at a clock. Of course, a screen opens up a range of new possible interactions. 'Barely crossed the starting line' Dominating this area isn't just about selling assistants. The opportunity for Amazon here is in an arena few thought they become a major player - home automation. Emarketer's data suggests that once you opt for one brand of assistant in your home, you're very unlikely to jump ship. So when the "internet of things" boom finally hits (any day now, as we've been saying the past three years) Amazon's early lead could really start to pay off. Or, it could blow it. Consider Amazon's lead like doing well in the first event of a heptathlon. "Amazon has a head start in the voice race but the industry has barely crossed the starting line," said CCS Insight analyst Geoff Blaber. I caught him as he was on his way to Microsoft's developer's conference, where its own digital assistant, Cortana, will be centre stage. He added: "Those that can maximize customer data, search, artificial intelligence and natural language processing, make it all available to developers to innovate with, and simultaneously walk the privacy tightrope, will be the ultimate winners." As it seeks to rapidly expand its lead, Amazon has made itself incredibly developer-friendly compared to its rivals.

I recently had a spin in an Alexa-enabled Ford, and General Electric today announced an Alexa-powered lamp. Amazon wants Alexa in as many nooks and crannies of our lives as possible.

Detected Text

Amazon surprised everyone when. in late 2014. it unveiled a standalone diaitat assistant that was not only good. but blew away the competition in both quality and aesthetics. The ticho - a eytindrical speaker with microphone - now accounts for just over 70%6 of all Jurital ausistant use in the US, leaving its nearest competitor, Google Home, well behind. #'s an important new market, even if the idea of talking to an object in your home still comes uneasity to many of us in a new report, Emarketer estimated 36 million Americans will use a voice-activated at last once a month - an increase of 129% on this time last year Amazon, as I mentioned. already has the lion's share, It's now hoping to echo (sorry) that with its latest effort which we could see as early as Tuesday. according to reports AIPTV.com, a site with a soild track record of leaks, said it found a low-quality image of the device on Amazon's own servers. The authenticity of the image was later backed up by king-of the- leaks. Evan "The new device is expected to house a 7-inch touchscreen and can be used for video calling. as well as displaying weather information and other data: It will help plug that wap that many voice assistant users will be familiar with. like not knowing how long a timer has left without asking. Or Just knowing the time -it's a step backwards to not just look at a clack. Of course. a screen opens up a range of new possible interactions. "tarely croused the starting tine" Dominating this area isn't just about selling assistants, The opportunity for Amazon here is in an arena few thought they became a major player - home automation. Emarketer's data sumrests that once you opt for one brand of assistant in your home. you're very unlikely to Jurnp ship. So when the "internet of things" boom finally hits (any day nowe as we've been saying the past three years) Amazon's early lead could really start to pay off. Or it could blow it. Consider Amazon's lead like doing well in the first event of a heptathlon. "Amazon has a head start in the vaice race but the industry has barely crossed the starting line." said CCS Insight analyst Geoff caught him as he was on his way to Microsoft's (developer's conference. where its own digital assistant, Cortana, will be centre stage: He added: "Those that can maximize customer data, yearch, artificial intelligence and natural language processing. make it all available to developers to innovate with. and simultancousty walk the privacy tightrope. will be the ultimate winners." As it seeks to rapidly expand its ead. Amazon has made itself incredibly developer-triendy compared to its rivals I recently had 'a spin in an Alexa-enabled Ford. and General Electric today announced an Alexa-powered lamp. Amazon wants Alexa in as many nooks and crannies of our tives as possible. t

13.1.2 Dubai becomes first city to get its own Microsoft font

http://www.bbc.com/news/business-39767990

Original Text

Not content with having the world's tallest building and biggest shopping centre, Dubai has become the first city to get its own Microsoft-designed font. The typeface comes in both Latin and Arabic script, and will be available in 23 languages. Government bodies have been told to use it in official correspondence. But given the human rights record of Dubai and the United Arab Emirates, eyebrows will be raised at claims it is a font of "self-expression". 'Create harmony' Dubai's Crown Prince Hamdan bin Mohammed al-Maktoum said he had been personally involved in "all the stages" of the development of the font. It was "a very important step for us as part of our continuous efforts to be ranked first in the digital world," he added. "We are confident that this new font and its unique specifications will prove popular among other fonts used online and in smart technologies across the world". Dubai's government

said the typeface's design "reflects modernity and is inspired by the city" and "was designed to create harmony between Latin and Arabic". When self expression isn't usually your type "Self-expression is an art form," says the blurb accompanying the launch of this font. "Through it you share who you are, what you think and how you feel to the world. To do so you need a medium capable of capturing the nuances of everything you have to say. "The Dubai Font does exactly that. It is a new global medium for self-expression." But the United Arab Emirates - of which Dubai is part - has been criticised for its restrictions on free speech. The constitution does guarantee the right to freedom of opinion and expression, but Human Rights Watch (HRW) says this "has no effect on the daily life of the citizen" and the country "has seen a wave of arrests and violations of human rights and freedoms and mute the voices of dissent". In March, high-profile human rights activist Ahmed Mansoor was arrested, a move HRW said showed "complete intolerance of peaceful dissent". The UAE's official news agency, WAM, said Mr Mansoor had been held "on suspicion of using social media sites to publish "flawed information" and "false news" to "incite sectarian strife and hatred" and "harm the reputation of the state."

Detected Text

Not content with having the world i tallest building and best shopping centre, Dubai has become the first city to get its own Microsoft designed font: "The typeface comes in both Latin and Arabic script. and will be available in 23 languages. Government bodies have been told to use it in official correspondence Wat aiven the human rights record of Dubai and the United Arab Emirates. eyebrows will be raised at claim it in a font of "welt-expression". 'Create harmony Bubal's Grown Prince Hamdan bin Mohammed al- Maktoum said he had been personally involved in "all the stages" of the development of the font. It was "a very important step for us as part of our continuous efforts to be ranked first in the world he added. "We are confident that this new font and its unique specifications will prove popular among other fonts used ontine and in smart technologies across the world". Dubai's government said the typeface's destin "reflects modernity and is inspired by the vity" and "was destined to create harmony between Latin and Arabic'. When self expression isn't usually your type "Seit exnression is an art form" says the blurb accompanying the launch of this font. "Ihrough it you share who you are. what you think and how you feel to the world. To do so you need a medium capable of capturing the nuances of everything you have to say. "Ime Dubai Font does exactly that. it is a new global medium for self-expression." hat the United Arab Emirates - of which Duba is part - has been criticised for its restrictions on tree speech "The constitution does nuarantee the right to freedom of opinion and expression. but Human Wights Watch (HRW) says this "has no effect on the daily life of the citizen" and the country "has ween a wave af arrests and violations of human rights and freedoms and mute the voices of dissent n March: high profile human rights activist Ahmed Mansoor was arrested, a move HRW said showed "compete intolerance of peaceful dissent ; The UAE's official news agency: WAM, maid Mr Manzoor had been held "on suspicion of using social media sites to publish "flawed information" and "false news" to "incite sectarian strife and hatred" and "harm the reputation of the state."

13.1.3 FCC website 'targeted by attack' after John Oliver comments

http://www.bbc.com/news/technology-39855490

Original Text

The US Federal Communications Commission (FCC) website was deliberately attacked on 8 May, the regulator has said. The incident began hours after comedian John Oliver criticised FCC plans to reverse US net neutrality rules. Mr Oliver urged people to post to the site's online commenting

system, protesting against the proposals. The FCC said that issues with the site were caused by orchestrated attacks, not high volumes of traffic. "These actors were not attempting to file comments themselves; rather they made it difficult for legitimate commenters to access and file with the FCC," chief information officer Dr David Bray said in an official statement. "While the comment system remained up and running the entire time, these distributed denial of service (DDoS) events tied up the servers and prevented them from responding to people attempting to submit comments." 'Trolling the trolls' In his Sunday night show Last Week Tonight, Mr Oliver called on viewers to visit a website that would direct them to the correct page on the FCC site to leave their comments. "Every internet group needs to come together gamers, YouTube celebrities, Instagram models, Tom from MySpace if you're still alive. We need all of you," he said. His plea came after FCC chairman Ajit Pai said in April that he would review rules made in 2015 that require broadband companies to treat all online traffic equally. Media captionEXPLAINED: What is a DDoS attack? Last December, Mr Pai said in a speech that the net neutrality laws were "holding back investment, innovation, and job creation". "Mr Pai is essentially trolling the trolls," Chris Marsden, professor of internet law at the University of Sussex, told the BBC. "If you bait John Oliver, you reap what you sow." The FCC will vote on Mr Pai's proposals to revoke the legislation on 18 May.

Detected Text

The US Fedterai Communications Commission (PCC) website was deliberately attacked on A May: the regulator has sas The incident began hours aer comedian John Oliver eriticised PCC plans to reverse US net rutes Mr Oliver urged people to post to the site's online commenting system. protesting against the nroposais The Fol said that issues with the ite were caused by orchestrated attacks, not high votumes of wathic "These actors were not attempting to file comments themselves; rather they made it for commenters to access and fle with the PCC. chief information officer Br David tay said in an official statement "While the comment system remained up and running the entire time, these distributed denial of service (DDoS) vents tied up the servers and prevented them fram responding to people attempting to submit comments" "Trotting the trots in his Sunday nught show Last Week Tonight Mr Oliver called on viewers to visit a website that would direct them to the carrect page on the PCC site to leave their comments "Avery internet group needs to come together... gamers. YouTube celebrities. Instagram models Tom from MySpace if you're sul alive We need all of you" he saic. His pes came aer FCC chairman Allt Pat sald in April that he would review rules made in 2018 Oiat require broadband companies to treat all onine traffic equally Media captionEXPLAINED: What is a DDoS attacker Last December. Mr Pas said in a speech that the net neutrality laws were "holding back Investment, innovation. and job creation", "hir Pal is eanentially trolling the trolls" Chris Marsden. professor of internet law at the University of Sussex. told the fie. "if you bait John Oliver. you reap what you sou" "The FCC will vote on Mr Pais proposals to revoke the legislation on 11 May.

13.2 Code

13.2.1 Raspberry Pi

13.2.2 Smart Phone Application

13.2.3 Server

Text Extraction

```
//
//  blurDetection.cpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: module to check whether the image (Mat object) received is blur

#include <opencv2/imgproc/imgproc.hpp>
#include "blurDetection.hpp"

//  threshold value to determine if the image is blur
#define BLUR_THRESHOLD 50

//  Function: varianceOfLaplacian
//  Description: generate the variance of Laplacian for the matrix received
double varianceOfLaplacian(cv::Mat &imageGray)
{
  cv::Mat laplacian_result;
  cv::Scalar mean;
  cv::Scalar stddev;

  Laplacian(imageGray, laplacian_result, CV_64F);
  meanStdDev(laplacian_result, mean, stddev);

  return pow((double) stddev[0],2);
}

//  Function: isBlur
//  Description: determine whether image received is blur or not
//    If the variance of Laplacian of the grayscalled image is less than the
    threshold
//    Then the image is blurred
bool isBlur(cv::Mat &image)
{
  cv::Mat imageGray;
  double variance;

  cvtColor(image, imageGray, cv::COLOR_BGR2GRAY);
  variance = varianceOfLaplacian(imageGray);

  if(variance < BLUR_THRESHOLD)
  {
    return true;
  }
  return false;
}
```

```
//
//  blurDetection.hpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: header file for blurDetection

#ifndef blurDetection_hpp
#define blurDetection_hpp

#include <opencv2/opencv.hpp>

bool isBlur(cv::Mat &image);

#endif /* blurDetection_hpp */
```

```
//
//  getImage.cpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: module for get the image for PRAHVI
//

#include "getImage.hpp"
#include "global.hpp"

//  Function: getImage()
//  Description: function that returns an opencv Mat object - an image for PRAHVI
    to process
//    Initially setup to read from a file, need to change with ios
//    TODO
cv::Mat getImage()
{
  cv::Mat image = cv::imread("/Users/Youngestyle/Desktop/image-19.jpeg");//
    fileAddress);
  return image;
}
```

```
//
//  getImage.hpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: header file for get the image for PRAHVI

```

```
#ifndef getImage_hpp
#define getImage_hpp

#include <opencv2/opencv.hpp>

cv::Mat getImage();

#endif /* getImage_hpp */
```

```
//
//  global.hpp
//  prahvi
//
//  Created by Yang Li on 5/6/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//

#ifndef global_hpp
#define global_hpp

extern std::string fileAddress;

#endif /* global_hpp */
```

```
//
//  imageToText.cpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: module that converts the image received to a string of text
//    the image received is alread preprocessed
//    currently just passes the image to the google tesseract api

#include <tesseract/baseapi.h>
#include "imageToText.hpp"


//  Function: replaceString
//  Description: replace all "toReplace" with "replaceWith" in string "s"
std::string replaceString(std::string &text, const std::string &toReplace, const
    std::string &replaceWith)
{
  int location = 0;
  int replaceWithLength = replaceWith.length();

  while((location = (int) text.find(toReplace, location)) != std::string::npos)
  {
    text.replace(text.find(toReplace), toReplace.length(), replaceWith);
    location += replaceWithLength;
  }
  return text;
}
```

```
//  Function: replaceLigatures
//  Description: replace the ligatures with non-ligatures
std::string replaceLigatures(std::string text)
{
  // list of ligatures and non ligatures
  // the list is too long, and it is making the system really slow
  //std::vector<std::string> ligatures = {"  ", "  ", "  ", "  ", "  ",
  //  "  ", "  ", "  ", "  ", "  ",
  //  "  ", "  ", "  ", "  ", "  ",
  //  "  ", "  ", "  ", "  ", "  ",
  //  "  ", "  ", "  ", "  ", "  ",
  //  "  ", "  ", "  ", "  ", "  ",
  //  "  ", "  "};
  //std::vector<std::string> nonLigatures = {"AA", "aa", "AE", "ae", "AO",
  //  "ao", "AU", "au", "AV", "av",
  //  "AV", "av", "AY", "ay", "ff",
  //  "ffi", "ffl", "fi", "fl", "OE",
  //  "oe", "OO", "oo", "fs", "fz",
  //  "st", "ft", "TZ", "tz", "ue",
  //  "VY", "vy"};

  // thus a shorter list of common ligatures are searched and replaced
  std::vector<std::string> ligatures = {"  ", "  ", "  ", "  ", "  ","  ", "
      "};
  std::vector<std::string> nonLigatures = {"ff", "ffi", "ffl", "fi", "fl", "st", "
    ft"};

  for(int i = 0; i < ligatures.size(); i++)
  {
    text = replaceString(text, ligatures[i], nonLigatures[i]);
  }

  return text;
}

//  Function: imageToText
//  Description: receive a Mat and pass the Mat to OCR to detect the text
//    The border of the image (Mat) is removed to reduce noise
//    The OCR is initialized for English ONLY.

std::string imageToText(cv::Mat &image)
{
  std::string outText;

  tesseract::TessBaseAPI *api = new tesseract::TessBaseAPI();

  // Initialize tesseract-ocr with English, without specifying tessdata path
  if (api->Init(NULL, "eng"))
  {
    std::cerr << "ERROR: could not initialize tesseract" << std::endl;
    exit(1);
  }

  // crop the image to remove the border
  // this reduces the noise from the background
  // can use fixed pixels or with respect to width and height

  int offsetX = image.size().width*0.05;
  int offsetY = image.size().height*0.05;

  cv::Rect roi;
  roi.x = offsetX;
```

```
91    roi.y = offsetY;
      roi.width = image.size().width - (offsetX*2);
93    roi.height = image.size().height - (offsetY*2);

95    // crop the original image to the defined ROI

97    image = image(roi);

99    // send the image to OCR
      api->SetImage((uchar*)image.data,
101             image.size().width,
                image.size().height,
103             image.channels(),
                image.step1());
105
      // get OCR result
107   api->Recognize(0);
      outText = api->GetUTF8Text();
109
      // destroy used object and release memory
111   api->End();

113   outText = replaceLigatures(outText);

115   return outText;
  }
```

```
1   //
    //  imageToText.hpp
3   //  prahvi
    //
5   //  Created by Yang Li on 4/29/17.
    //  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
        All rights reserved.
7   //
    //  Description: header file for imageToText
9
    #ifndef imageToText_hpp
11  #define imageToText_hpp

13  #include <opencv2/opencv.hpp>

15  std::string imageToText(cv::Mat &image);

17  #endif /* imageToText_hpp */
```

```
    //
2   //  main.cpp
    //  prahvi
4   //
    //  Created by Yang Li on 4/29/17.
6   //  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
        All rights reserved.
    //
8   //  Description: the system test file after the image is received
    //    this file creates the prahvi object and convert the image to text
10
```

```
#include <iostream>
#include "prahvi.hpp"
#include "global.hpp"

std::string fileAddress;

int main(int argc, const char * argv[]) {

  int result;
  std::string text;

  /*
   if(argc < 2)
  {
    std::cerr << "ERROR: file name not specified" << std::endl;
    return -1;
  }

  fileAddress = argv[1];
   */

  prahvi myPrahvi;
  text = myPrahvi.getNewText(result);
  if(result == SUCCESS)
  {
    std::cout << text << std::endl;
  }
  else if(result == EMPTY)
  {
    std::cout << "Empty image, try again" << std::endl;
  }

  return 0;
}
```

```
//
//  prahvi.cpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: prahvi class
//    the prahvi class does the preprocessing and text detection
//    other program can create and call this class to get corresponding results

#include "prahvi.hpp"
#include "getImage.hpp"
#include "blurDetection.hpp"
#include "similarityDetection.hpp"
#include "imageToText.hpp"
#include "scanner.hpp"
#include "boundingBoxDetection.hpp"

//  Function: prahvi::prahvi
//  Description: constructor for prahvi
prahvi::prahvi()
```

```
{
  _previousImage = cv::Mat::zeros(1, 1, CV_64F);
  _currentText = "";
  _currentImage = cv::Mat::zeros(1, 1, CV_64F);
}

//  Function: prahvi::getText
//  Description: get the text of the current image
std::string prahvi::getText()
{
  return _currentText;
}

//  Function: prahvi::getNewText
//  Description: get a new image and process it
//    the fucntion will get a new image
//    if the new image is blur, it will terminate
//    otherwise, it will extract the text area
//    and compare to the previous text area
std::string prahvi::getNewText(int &result)
{
  cv::Mat newImage = getImage();

  //  check if the new image is blurred
  if(isBlur(newImage))
  {
    result = BLUR;
    return "";
  }

  _previousImage = _currentImage;
  _currentImage = getTextArea(newImage);

  //  check if the new image is similar to the previous image
  //  TODO - uncomment after add IDF
  /*
  if(_previousImage == cv::Mat::zeros(1, 1, CV_64F) || isSimilar(_previousImage,
    _currentImage))
  {
    result = SIMILAR;
    return "";
  }
   */

  //  convert the image to text
  _currentText = imageToText(_currentImage);

  //  reset TF-IDF and generate the score for the new document
  //  TODO - uncomment after add IDF
  //_tfidf.resetTerms();
  //_tfidf.addTerms(_currentText);

  result = EMPTY;

  for(int i = 0; i < _currentText.length(); i++)
  {
    if(!isspace(_currentText[i]))
    {
      result = SUCCESS;
    }
  }
```

```
  return _currentText;
}

std::string prahvi::getKeyword(int n)
{
  //  TODO - uncomment after add IDF
  return "";//_tfidf.getTerm(n);
}
```

```
//
//  prahvi.hpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: header file for prahvi class

#ifndef prahvi_hpp
#define prahvi_hpp

#include <opencv2/opencv.hpp>
#include "tfidf.hpp"

enum ProcessResult {SUCCESS, BLUR, SIMILAR, EMPTY};

class prahvi
{
public:
  prahvi();
  std::string getText();
  std::string getKeyword(int n=1);
  std::string getNewText(int &result);

private:
  cv::Mat _previousImage;
  cv::Mat _currentImage;
  std::string _currentText;
  //  TODO - uncomment after add IDF
  //tfidf _tfidf;
};

#endif /* prahvi_hpp */
```

```
//
//  scanner.cpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: module that extract the text area from the image
//    such that the result will be like a scanned document

#include <algorithm>
```

```cpp
#include <vector>
#include "scanner.hpp"

//  Function: comaprePointSum
//  Description: compare 2 points based on the sum of the coordinate
//    return true if the first point is smaller than the second point
bool comparePointSum(cv::Point a, cv::Point b)
{
  return a.x + a.y < b.x + b.y;
}
//  Function: comaprePointDifference
//  Description: compare 2 points based on the difference of the coordinate
//    return true if the first point is smaller than the second point
bool comparePointDifference(cv::Point a, cv::Point b)
{
  return a.y - a.x < b.y - b.x;
}

//  Function: compareArea
//  Description: compare 2 points based on the contor area
//    return true if the first point is larger than the second point
bool compareArea(std::vector<cv::Point> a, std::vector<cv::Point> b)
{
  return contourArea(a) > contourArea(b);
}

//  Function: getDistance
//  Description: return the distance between two points
int getDistance(cv::Point a, cv::Point b)
{
  return sqrt(pow((double)b.x - (double)a.x, 2) + pow((double)b.y - (double)a.y, 2)
    );
}

//  Function: sortContours
//  Description: sort the contours based on the contour area
//    in descending order
void sortContours(std::vector<std::vector<cv::Point>> &contours)
{
  sort(contours.begin(), contours.end(), compareArea);
}

//  Function: getTextArea
//  Description: extract the text area from the image
//    Based on find the largest contour with 4 sides in the image
//    this function also transform the result found and rectify it
cv::Mat getTextArea(cv::Mat &image)
{
  // convert to grayscale and blur
  image.convertTo(image, -1, 1, 20);
  cv::Mat imageGray;
  cvtColor(image, imageGray, CV_BGR2GRAY);

  cv::Mat blurred;
  GaussianBlur(imageGray, blurred, cv::Size(5, 5), 0);

  // apply Canny Edge Detection to find the edges
  cv::Mat edged;
  Canny(blurred, edged, 0, 50);

  //  find the contours in the edged image
  std::vector<std::vector<cv::Point>> contours;
```

```
73  std::vector<cv::Vec4i> hierarchy;

75  findContours(edged, contours, hierarchy, cv::RETR_LIST, cv::CHAIN_APPROX_NONE);

77  //  sort the contours in descending order
    sortContours(contours);
79
    //  initialize the screen contour
81  std::vector<cv::Point> screenContour;
    std::vector<cv::Point> approx;
83
    //  set screen contour to the largest contour with 4 sides
85  for(int i = 0; i < contours.size(); i++)
    {
87    double peri = arcLength(contours[i], true);

89    approxPolyDP(cv::Mat(contours[i]), approx, 0.02*peri,true);

91    if(approx.size() == 4)
      {
93      screenContour = approx;
        break;
95    }
    }
97
    std::vector<std::vector<cv::Point>> screen;
99  screen.push_back(screenContour);

101 //  initialize transformation
    cv::Mat lambda(2, 4, CV_32FC1);
103 lambda = cv::Mat::zeros(image.rows, image.cols, image.type());

105 //  input and output coordinates
    cv::Point2f inputQuad[4];
107 cv::Point2f outputQuad[4];

109 //  find the max dimension of the crop
    cv::Point topLeft, topRight, bottomRight, bottomLeft;
111
    //  the top left point has the smallest sum
113 topLeft = *min_element(screenContour.begin(), screenContour.end(),
      comparePointSum);

115 //  the bottom right point has the largest sum
    bottomRight = *max_element(screenContour.begin(), screenContour.end(),
      comparePointSum);
117
    //  the top right point has the smallest difference
119 topRight = *min_element(screenContour.begin(), screenContour.end(),
      comparePointDifference);

121 //  the bottom left point has the largest difference
    bottomLeft = *max_element(screenContour.begin(), screenContour.end(),
      comparePointDifference);
123
    //  set input coordinates
125 inputQuad[0] = topLeft;
    inputQuad[1] = topRight;
127 inputQuad[2] = bottomRight;
    inputQuad[3] = bottomLeft;
129
    //  the dimension of the output is based on the input
```

```
131     //  1:1 ratio
        int width = std::max(getDistance(topLeft, topRight), getDistance(bottomLeft,
          bottomRight));
133     int height = std::max(getDistance(topLeft, bottomLeft), getDistance(topRight,
          bottomRight));

135     //  the output coordinates is based on the output dimention
        outputQuad[0] = cv::Point2f(0,0);
137     outputQuad[1] = cv::Point2f(width-1, 0);
        outputQuad[2] = cv::Point2f(width-1, height-1);
139     outputQuad[3] = cv::Point2f(0, height-1);

141     //  set up transformation
        lambda = getPerspectiveTransform(inputQuad, outputQuad);
143
        cv::Mat output;
145
        //  apply transformation
147     warpPerspective(image, output, lambda, cv::Size(width,height));

149     return output;
    }
```

```
1   //
    //  scanner.hpp
3   //  prahvi
    //
5   //  Created by Yang Li on 4/29/17.
    //  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
        All rights reserved.
7   //
    //  Description: header file for the scanner module
9
    #ifndef scanner_hpp
11  #define scanner_hpp

13  #include <opencv2/opencv.hpp>

15  cv::Mat getTextArea(cv::Mat &image);

17  #endif /* scanner_hpp */
```

```
    //
2   //  similarityDetection.cpp
    //  prahvi
4   //
    //  Created by Yang Li on 4/29/17.
6   //  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
        All rights reserved.
    //
8   //  Description: module to detect whether the two image received are similar
    //    Currently, the method used is AKAZE tracking
10  //    Can be improved using matching


12


14  #include <opencv2/features2d.hpp>
```

```
#include <opencv2/imgcodecs.hpp>
#include <vector>
#include "similarityDetection.hpp"

//  threshold value to determine whether the two images are similar
#define FEATURE_THRESHOLD 1000


const float inlier_threshold = 2.5f; // Distance threshold to identify inliers
const float nn_match_ratio = 0.8f;   // Nearest neighbor matching ratio

//  Function: akazeTracking
//  Description: comapre two images and return the number of good match points
//    Uses the A-Kaze tracking
int akazeTracking(cv::Mat &image1, cv::Mat &image2)
{
  // convert the images to grayscale
  cv::Mat image1Gray;
  cv::Mat image2Gray;

  cvtColor(image1, image1Gray, cv::COLOR_BGR2GRAY);
  cvtColor(image2, image2Gray, cv::COLOR_BGR2GRAY);


  // detect keypoints and compute descriptors using A-KAZE
  std::vector<cv::KeyPoint> keyPoints1, keyPoints2;
  cv::Mat descriptors1, descriptors2;

  cv::Ptr<cv::AKAZE> akaze = cv::AKAZE::create();
  akaze->detectAndCompute(image1Gray, cv::noArray(), keyPoints1, descriptors1);
  akaze->detectAndCompute(image2Gray, cv::noArray(), keyPoints2, descriptors2);

  // use the brute force matcher to find 2-nn matches
  cv::BFMatcher matcher(cv::NORM_HAMMING);
  std::vector<std::vector<cv::DMatch>> nn_matches;
  matcher.knnMatch(descriptors1, descriptors2, nn_matches, 2);

  // if one or more of the image does not have any keypoint, return 0
  if(keyPoints1.size() <= 0 || keyPoints2.size() <= 0)
  {
    return 0;
  }

  // use 2-nn matches to find correct keypoint matches
  std::vector<cv::KeyPoint> matched1, matched2, inliers1, inliers2;
  std::vector<cv::DMatch> good_matches;

  for(size_t i = 0; i < nn_matches.size(); i++)
  {
    cv::DMatch first = nn_matches[i][0];

    float distance1 = nn_matches[i][0].distance;
    float distance2 = nn_matches[i][1].distance;

    if(distance1 < nn_match_ratio * distance2)
    {
      matched1.push_back(keyPoints1[first.queryIdx]);
      matched2.push_back(keyPoints2[first.trainIdx]);
    }
  }

  // check if the matches is within the inlier_threshold
```

```
  for(unsigned i = 0; i < matched1.size(); i++) {
    cv::Mat col = cv::Mat::ones(3, 1, CV_64F);
    col.at<double>(0) = matched1[i].pt.x;
    col.at<double>(1) = matched1[i].pt.y;

    col /= col.at<double>(2);
    double distance = sqrt(
                pow(col.at<double>(0) - matched2[i].pt.x, 2)
                + pow(col.at<double>(1) - matched2[i].pt.y, 2)
                );

    if(distance < inlier_threshold) {
      int new_i = static_cast<int>(inliers1.size());
      inliers1.push_back(matched1[i]);
      inliers2.push_back(matched2[i]);
      good_matches.push_back(cv::DMatch(new_i, new_i, 0));
    }
  }
  return good_matches.size();
}

bool isSimilar(cv::Mat &image1, cv::Mat &image2)
{
  if(akazeTracking(image1, image2) > FEATURE_THRESHOLD)
  {
    return true;
  }
  return false;
}
```

```
//
//  similarityDetection.hpp
//  prahvi
//
//  Created by Yang Li on 4/29/17.
//  Copyright    2017 Portable Reading Assistant Headset for the Visually Impaired.
     All rights reserved.
//
//  Description: header file for similarityDetection

#ifndef similarityDetection_hpp
#define similarityDetection_hpp

#include <opencv2/opencv.hpp>
bool isSimilar(cv::Mat &img1, cv::Mat &img2);

#endif /* similarityDetection_hpp */
```