iRis: A Wearable Device for Reading Printed Text

A Project Report

Submitted By

ABEL SAJU AUGUSTINE (11EC05)

ASWIN SURESH KRISHNAN (11EC19)

DISHIT DAK (11EC31)

SHARON SANJEEV GEORGE (11EC88)

under the guidance of

Dr. RAMESH KINI M.

Associate Professor

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April 15, 2015

ABSTRACT

Visually Impaired (VI) people find it difficult to access information in the form of printed text. Existing solutions in the market these days make tradeoffs between processing speed, size and efficiency. Using an intelligent OCR-based algorithm coupled with real-time auditory and tactile feedback, iRis provides the user with an assistive text-reading experience for printed text. The device can also perform translation between languages and provide word definitions of the captured text. iRis is designed in such a way so as to permit easy setup-and-use with minimum calibration and high mobility. The prime motivation for this design can be attributed to the preliminary studies conducted with visually impaired people.

Place:NITK, Surathkal

Date:15/04/2015 Signature of the Mentor:

Contents

List of Figures

1 Introduction

1.1 Problem Definition

To design a device that can perform real-time character recognition on printed text and convert the obtained input into audio signals. If successful, this can provide a solution to the aforementioned problem. This could alleviate the restrictions regarding form factor of the printed material that deter the blind from accessing information.

1.2 Previous Work

iRis is a wearable device that detects printed text through word detection in real time with respect to the finger position. We have introduced guidance signals which helps users to move up, down and to the next line. In addition to this, it also provides features like word definition and translation between inbuilt language packs.

Although MIT FingerReader [1] and OrCam [2] have a real time point and read word detection system most others do not. For instance, Zoom-Twix [3] scans the page and then reads the content and thus does not provide instant feedback and the kNFB K-Reader [4] works by taking a snapshot of a text passage which is then processed. Most of the devices do not have any guidance signals . Moreover, the MIT FingerReader [1] does not have the guidance mechanism which aids the user to navigate to next upon reaching the end of a line. Amongst the text readers compared none of the devices support word definition, and k-Reader Mobile [4] is the only device that can translate between languages. However, JAWS [5], Eye-Pal ROL [6] and Zoom-Twix [3] are available in various world languages. Most devices are portable and can read printed text. However, JAWS [5] is a desktop based application and can only read on-screen text.

1.3 Motivation

The stimulus for this project came as result of a desire to take the theoretical knowledge learned in classrooms onto a practical scenario, in doing so contribute to the society. During our preliminary research, we found this to be an area which has not seen much work until recently. India is currently home to the largest population of blind people and we account for 20% of the world's blind. Hence, we aim to produce a viable solution to a highly neglected problem.

1.4 Overview

With this project, our aim is to use technology to bridge the gap that deter the visually impaired from accessing information. Therefore, we present our work from the past 9 months in tackling this issue by building a mobile device, iRis, that will allow visually impaired people to read text off printed materials in real-time. Some of the features of iRis is as follows:
1. A point-to-read input mechanism which performs local sequential scanning of text with real-time auditory and tactile feedback. This allows the user to read any section of the text without any restrictions.

2. iRis employs assistive text-reading techniques which helps the user keep track of where reading is performed.

3. Besides text reading, the device presents the user with options to perform some other functions:
   (a) The translation of words from one language to the other.
   (b) Use of an English dictionary to find word definitions.

The prototype is mountable on a finger with all the necessary modules assembled into a single apparatus.

2 Description

Software

We developed a software that consists of algorithms for text extraction and real-time image processing. It is integrated with Tesseract OCR and Python Text-To-Speech (TTS) and is done with the help of Open Source Computer Vision Library. The program supports three modes namely reading, translate and define. On start, the user is asked to choose the required mode of operation. The reading mode simply scans for text and provides a real-time auditory feedback. The translate mode provides a real-time translation of the detected text to any of the included language packs. An internet connection is mandatory for this mode to operate. The define mode provides the user with the definition of the scanned word along with all possible usages, exactly as in a dictionary.

2.1 Working Mechanism

The program begins with detecting the fingertip wherein the user is asked to hold the finger steady for a few seconds until the tip coordinates are acquired. These coordinates are later used for finding the region of interest (ROI) for text extraction. A close-up view of the printed text is required as input for text extraction. We scan the frame continuously for words and detect all the words in a given frame. Thereafter we look for words that lie in the ROI and extract them. These words are then sent to the OCR Engine. The recognized text undergoes a dictionary check in order to ensure validity. Repetitive detection is avoided. If the recognized word is found valid, we invoke the TTS engine to utter the word.

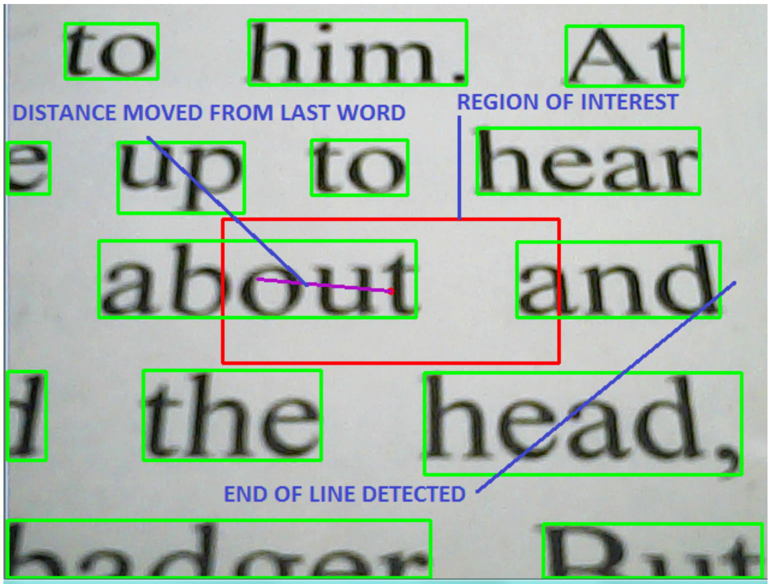

Figure 1: Software detecting words and End of Line while reading. Last extracted word: **about**

2.2 Word Detection

To the captured frame, we apply image binarization and blurring in order to get clear text. Then we calculate all the edges in the frame. Then, Hough Transform (HT) is [7] used to find lines through each edge pixel at every possible angle. These lines are carefully chosen in such a way that they overlap the letters of the same word and no line crossconnects two existing words in the text. The transformation between feature space and parameter space is done [8] during HT. For using HT, we rely on the fact that every point in the image space can lie on a line of the form r=xcos(Ø) +ysin(Ø) (where r is the length of a normal from the origin to the line and (Ø) is the orientation of r with respect to the x-axis) in the Hough parameter space. For every black pixel, we find lines along the range of angles specified and then increment the accumulator by one. We finally retrieve a resultant matrix which contains values indicating the number of points that lie on a particular line. Lines that contain more pixels will be attributed a higher value than those that contain fewer pixels. We set a lower threshold to discard lines with small values. Maximum distance between two points to be on the same line is also chosen exquisitely so that the length of each line is restricted. Once this is established, we find the contours and draw bounding figures over the detected contours for the upcoming processes of extraction and recognition. (**Fig 3: Process 1**)

2.3 Word recognition

The extracted word is resized and passed onto the Tesseract OCR for the recognition process (image-to-text conversion). The returned string undergoes a dictionary check. If a match is found, then the word is stacked up for text-to-speech conversion. (**Fig 3: Process 2**)

2.4 Text-to-Speech conversion

When new data is found in the stack, it is emptied and the text is passed on to the TTS engine for text-to-speech conversion and the word is uttered aloud. For transfer of data extracted to engine and between the two engines, we use a Last-In-First-Out (LIFO) approach by which we give higher priority to the most recently obtained data. If the program finds out that the user has moved ahead quickly and the earlier elements in the stack are of no use, then it is cleared and the process continues. (**Fig 3: Process 3**)

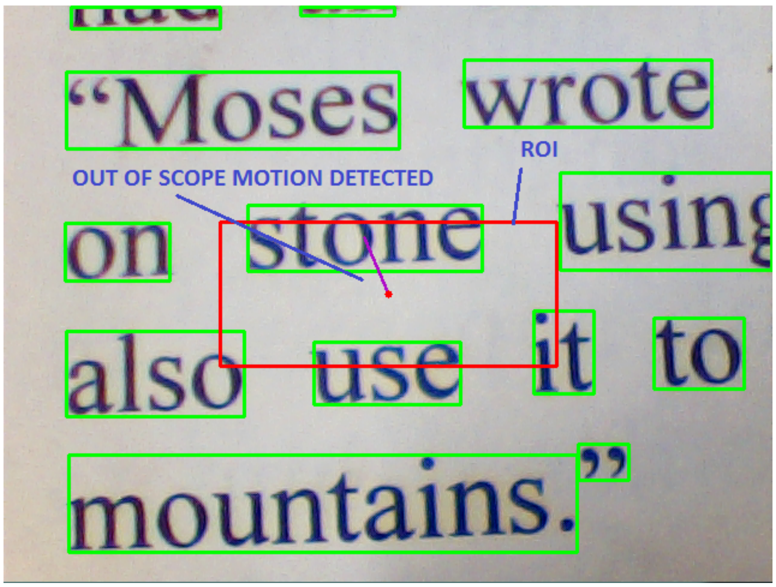

Figure 2: Accidental motion belowthe line. Warning to move upis triggered.

2.5 Finger guidance mechanism

We keep track of the position of the finger with respect to the current line of scan. If the user deviates to a position above or below, we trigger a tactile feedback, warning the user to return the finger to its correct path. When no more words are detected further ahead in the current line, an end of line warning is issued and the user is guided to move to the line just below. Furthermore, the user is expected to make a return path along this new line until the start of the line is reached. During this process, the word extraction is at a halt and will continue after the user has successfully managed to maneuver the finger to the start of the line. To ensure successful transfer of finger to the next line, we keep track of all the words below the reading line and use a nearest neighbour algorithm to find out the last word of the next line. We check for End of Line and Start of Line conditions by periodically searching for words to the left and right of the Region of Interest. If there are no more words to read in the current line, we trigger the End of Line condition. Also, the distance from the point of maximum interest (finger-tip position) to the center of the last detected word is continuously tracked to ensure the path of motion. If it is found that the user has deviated to a position significantly higher or lower than required with respected to the center of previous word, then it is assumed that the user has moved above or below the line of reading and a guidance mechanism is activated which helps the user to return to the line.

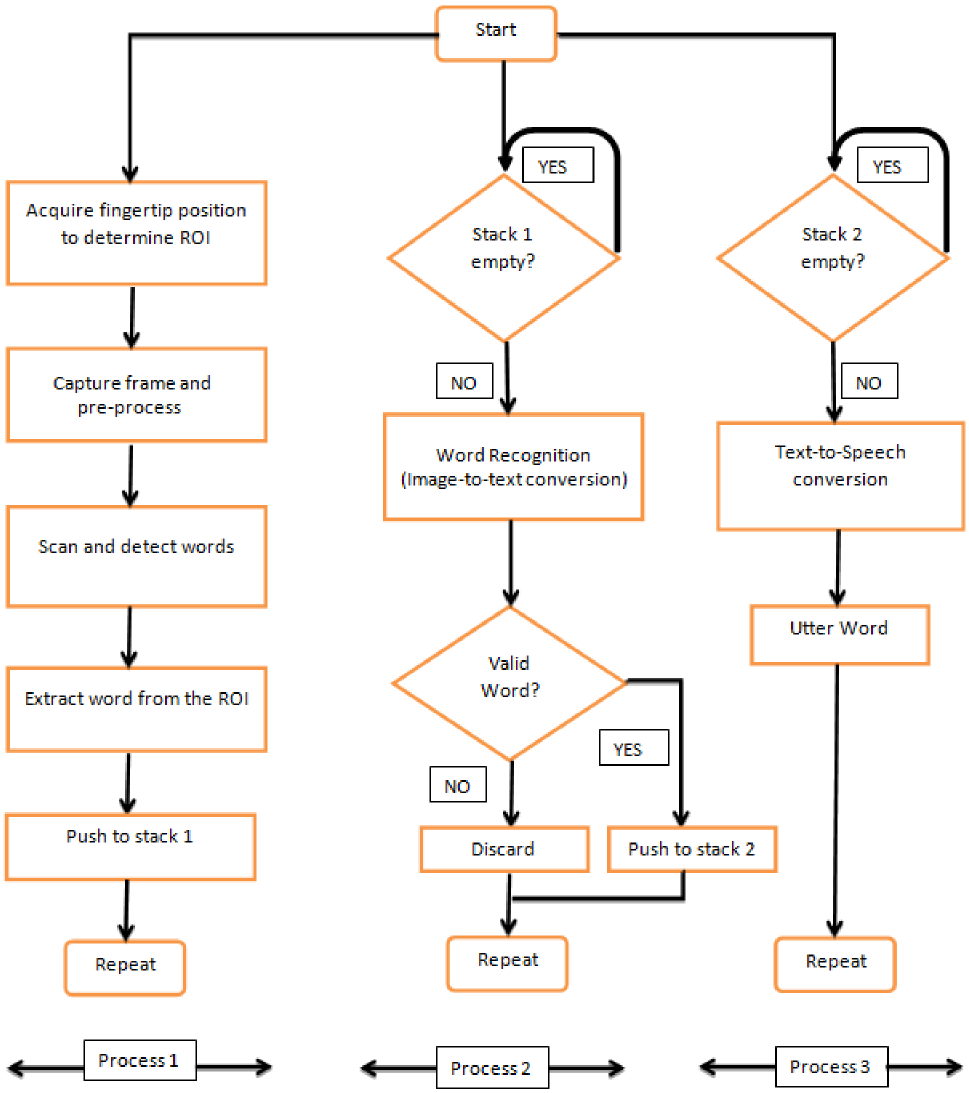

Figure 3: Text Reading Algorithm Flowchart

2.6 Compatability:

Our software runs on Windows and Mac machines with maximum efficiency. It also runs on Raspbian OS but with higher processing time. The main focus was on real-time processing with minimum running time. This is a necessary condition for real-time feedback. We made sure that the processes of word detection and extraction, image-to-text conversion and text-to-speech conversion take place simultaneously [9], thus reducing program lag and processing time.

2.7 Problems that were resolved

1. The huge delay problem was reduced significantly when parallel processing was employed.
2. The incorrect detection problem is reduced significantly by using trained files generated manually instead of using an existing OCR engine like Tesseract.
3. The program is able to detect camera motion using differential imaging. (Up to a level; threshold is yet to be set accurately for better results). This helped in avoiding detection of repeated letters.
4. A warning signal is successfully generated when the user moves out of required region (scope) and during the end of each line (so that the user can move to the next line).

2.8 Prototype Design

The 3D modeling of the product was done on a software called **CATIA** (Computer Aided Three-dimensional Interactive Application). One of the key features that was exploited was the one that allowed us to draw on a 2D plane (sketching) and then extend it to the 3D plane by adding depth or creating gaps.

iRis is a finger-worn device that assists visually challenged people in reading printed text. The device is based on ABS material. In the front view, we can see a small rectangle gap. This is the exposed section of the camera module. The circular hole at the bottom is where the finger is inserted. The small gap beneath the circular hole is to allow for fingers of larger sizes (within limitation).

In the back view, one can notice two slits. The top slit is for a "sliding gate" mechanism to close the open top. A slab of similar dimensions can be inserted through this gap and close off the top gap which is kept to insert the camera module, wiring and lights. The bottom slit is kept as an exit for the rpi camera's wire. The side view also shows a pocket that would allow an exit for the other wires.

The inside view shows the cavity where the camera module sits along with the other wires. The angled projection of the front side of the device allows the camera to be at a fixed position, capturing both the fingertip and the printed text in a single frame. It, initially, locks your finger position and thus allows you to point to the text that is to be read. The device captures the images through the Raspberry Pi cam which is used for the OCR. The processing was tested on desktops as well as the Raspberry Pi 2. The Raspberry Pi 2 allows us more portability and is a cheap alternative. The hardware model also contains high brightness LEDs to illuminate the print.

The feedback is provided through tactile and auditory stimuli [10]. Two vibration motors are present to provide tactile feedback when the user strays above or below the printed line. Auditory signals are emitted when the user reaches the end of line. The user is then guided to the new line and receives another auditory signal when he/she is on the new line. The third tone is emitted at the start of the new line. These five signals, two tactile and three auditory signals, act as guiding signals for the users.

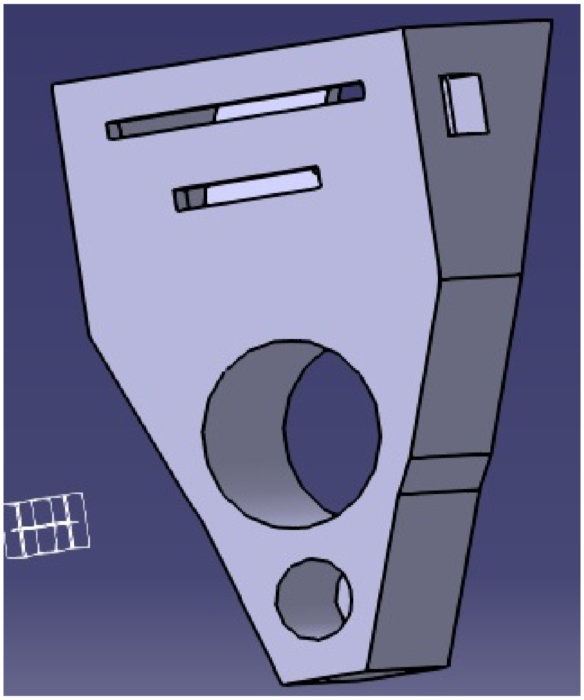

Figure 4: (i) Back View

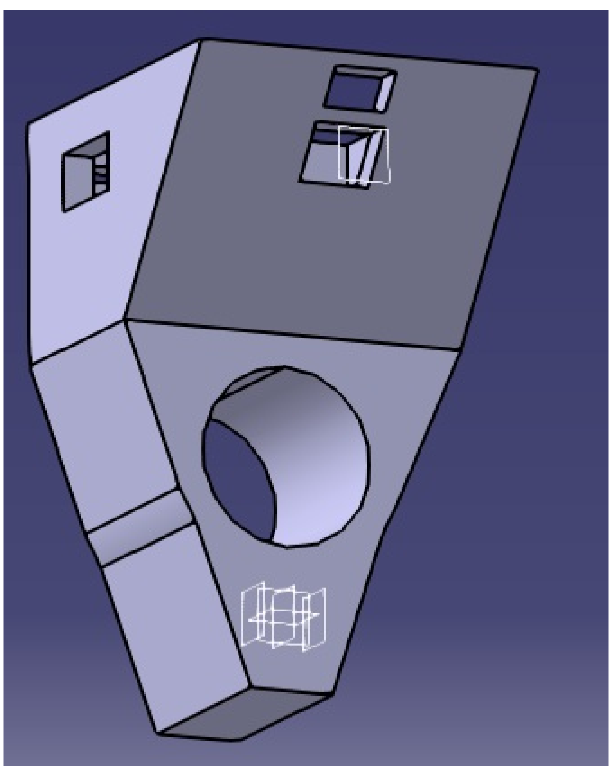

Figure 5: (ii) Front View

3 Conclusion

In this project, a vision based text-reader is developed for the visually impaired using the basic principles of Optical Character Recognition (OCR) along with settings that are user-friendly. The reader works on a real-time basis continuously scanning for text and converting it into speech. The project has helped us in getting a deeper insight into OCR as well as in the areas of object detection and resolution, image analysis, training, data file generation and protoyping. Hence, on integrating our algorithm with the respective hardware, a working prototype was implemented marking the successful completion of our project.

3.1 Future Work

The problems that were encountered and the challenges that we had to overcome, has helped us stay motivated and advance the project into a stage closer to completion. The following can be accomodated to make the device portable:

1. Portable MCU that has sufficient processing speed.
2. Currently the processing is primarily done on a computer. Work can be done on making the system more portable by finding ways to eliminate the need for a computer.
3. The algorithm was compiled and tested on Raspberry Pi 2 board but the processing speed was quite low. Since we are adopting a real-time approach, there can be no compromise on speed. Therefore, optimization of the algorithm on such hardware can be lucrative.
4. iRis works on text printed on paper. Work can be done to accommodate non-paper based domains.
5. Training of the OCR engine to work with all kinds of fonts of different sizes. This calls for the need to address the issue of camera focus as well, wherein the camera can focus automatically for the diverse fonts and domains.

References

[1] FingerReader:A Wearable Device to Support Text Reading on the Go, http://fluid.media.mit.edu/sites/default/files/paper317.pdf

[2] OrCam, 2013, http://www.orcam.com/.

[3] ABiSee's Zoom-Twix, 2007, http://www.abisee.com/products/zoom-twix.html

[4] KNFB kReader mobile, 2010, http://www.knfbreader.com/products-kreader-mobile.php.

[5] JAWS Screen Reader,2008, http://www.freedomscientific.com/Products/Blindness/JAWS.

[6] ABiSee's EyePal ROL, 2013, http://www.abisee.com/products/eye-pal-rol.html.

[7] A Hough Transform based Technique for Text Segmentation by Satadal Saha, Subhadip Basu, Mita Nasipuri and Dipak Kr. Basu.

[8] Multifont Arabic Character Recognition Using Hough Transform and Hidden Markov Models by Nadia Ben Amor and Najoua Essoukri Ben Amara.

[9] http://chimera.labs.oreilly.com/books/1230000000393/ch12.html , Python Cookbook , Chapter 12- Concurrency.

[10] FingerReader:A Wearable Device to Explore Printed Text on the Go, http://fluid.media.mit.edu/sites/default/files/FingerReaderCHI15