Computer Vision and Text Recognition for Assisting Visually Impaired People using Android Smartphone

Hao Jiang, Thomas Gonnot, Won-Jae Yi and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— Advances made with new technologies have boosted the development of systems to assist the daily lives of the visually impaired people. These systems intend to help by providing their user with some critical information about their environment using senses they can still use. In this paper, we discuss a system that uses existing technologies such as the Optical Character Recognition (OCR) and Text-to-Speech (TTS) available on an Android smartphone, and use them to automatically identify and recognize texts and signs in the environment and help the users navigate. The proposed system uses a combination of computer vision and Internet connectivity on an Android smartphone not only to recognize signs, but also reconstruct sentences and convert them to speech. This paper discusses the design flow and the experimental results of the project.**

I. Introduction

There is a significant amount of effort put into the modification of various infrastructures around the world to help the visually impaired people to interact with their environment. With an estimated 285 million visually impaired people in the world however, from which about 39 million are blind, it is difficult but also expensive to modify all the infrastructures available to them [1]. Simple solutions such as canes already helps the blind navigate their environment by avoiding most obstacles in their way [2], but it doesn't help then in the case they need to read the different directions signs and room numbers, office names and so on. Braille has been used for a long time now, and is available in most public places [3], but it is usually limited to fixed signs and doesn't extend to temporary posting. In the case, visually impaired people would need to read a text, Optical Characters Recognition devices exist and allow then to scan the text line by line and either convert it to braille or read it using Text-to-Speech algorithms [4]. However, these devices require a contact with the document being read, which implies knowing that the information is available, and having a physical access to it.

The proposed system tries to tackle these two issues. Using a camera, the idea is to automatically locate the different sources of information in the environment, and using Text-to-Speech techniques, inform the user of their location. The system also uses OCR to read the different sources and relate their content to the user using Text-to-Speech algorithms as well. In order to keep the system affordable, it is designed as an application on a smartphone. The advantages of smartphones over other devices is not only the fact that they are affordable, but also that they are very powerful, with most models nowadays integrating several cores in their main processor. They are also standalone devices with a camera, a battery, and audio output and an Internet connection.

In this paper, we first describe the system design using the Android smartphone. We then discuss the signs detection algorithm and the Optical Character Recognition engine. Finally, we discuss the feedback operation, before presenting the preliminary results of the system.

II. System Design

A. Requirements

The aim of this project is not only to reduce the visually impaired people dependency from the environment, but also from costly or bulky devices. Therefore, this system is designed to be deployed on a large number of smartphones. Practically every smartphone running Android that are being sold nowadays have at least one camera that can be used for the purpose of this project. The smartphone also comes with several features that can be used, such as an increasing number of processing units and even GPUs and DSPs that can process images faster than conventional processors, Internet connectivity through Wi-Fi or cellphone network, and multiple motion sensors. Finally, the smartphone all have an audio interface that can be used to convert the text to speech.

B. Design Flow

Figure 1 shows the design flow of the system from the image acquisition to the ear of the user. The flow is decomposed in two main processes, one process in in charge of the OCR, and to read the result to the user, and the other process keeps track of the signs and texts and generated a feedback to the user to turn the head in order to frame it correctly.

978-1-5090-4767-3/17/$31.00 ©2017 IEEE

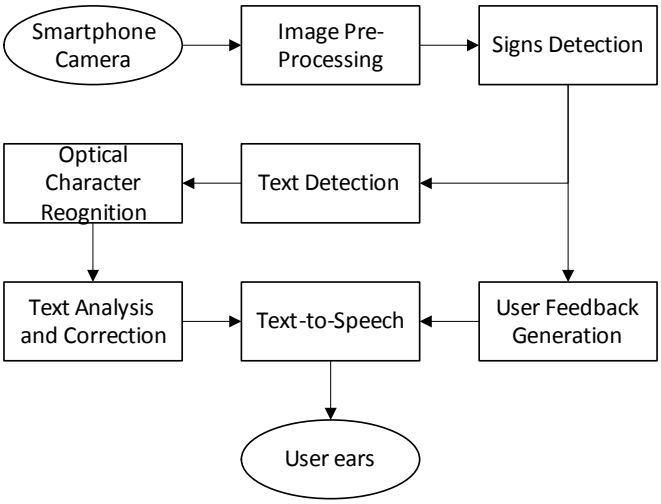

Figure 1. Design flow of the system

The text recognition process decomposes in three main steps. First the image is processed by the OCR algorithm, that inputs the region containing the sign with the text to be recognized, and returns a raw text. Then the text is forwarded to a second algorithm in charge of correcting the text and recovering sentences when possible. Finally, the result is converted to text using the Text-to-Speech algorithm.

The user feedback is also composed of three main steps. The first step consists in detecting the signs in the image. Once all the signs are located, they are forwarded to the OCR process, and to another algorithm that will evaluate how the camera should be moved to optimize the recognition of the signs. Finally, a set of commands are generated and sent to the Text-to-Speech algorithm to be communicated to the user.

III. Signs Detection

The main goal of the image pre-processing is to prepare the image for the sign detection. The goal is to be able to separate as much as possible the signs from the background using color filters or other techniques. Figure 2 shows a practical example of a scene with a complex environment. The image on the right highlights what would be the idea signs to be detected. Note that not all signs will have text, but this information could be used in future implementation of the project.

Figure 2. Ideal sign detection

The same concept can be applied to indoor navigation by trying to identify signs on the walls and doors. However, indoor navigation represents a challenge as there is usually no standard design for the signs used, and color extraction might be useless in some situations. Shape detection can help remove the issue in some case by focusing on the rectangular shapes expecting to catch a sign. In any case, the OCR algorithm can help discarding selections that are considered non-relevant, or without text.

IV. Optical Character Recognition

The system described in this paper is built around the OCR algorithm. Rather than creating a new OCR implementation, the open-source OCR engine called Tesseract was chosen [5]. It was developed by HP from 1985 until 1994, which opened its source in 2005. Since 2006, the engine is being developed and used by Google. However, several OCR engines uses similar techniques to "read" the text in an image.

First the text must be localized in the image. The algorithm needs to extract the layout of the text before separating the different words. In order to do that, it uses a line fitting that determines the different lines of text available, and their size and orientation, as show in Figure 3.

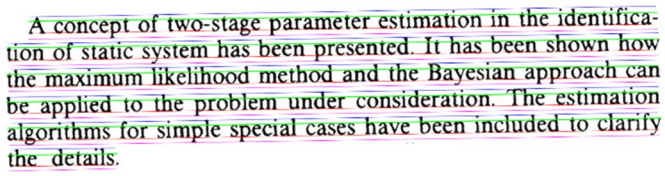

Figure 3. Sample text and layout detection

The next step consists in detect the spacing between the letter and find the spaces by analyzing the average space between the letters and their average size. From there two techniques can be used: with a fix pitch detection or the variable pitch. In the first case, the pitch is fixed to the average letter separation and the characters are separated using that pitch. However, a lot of fonts don't have a fixed pitch, and using this method would result in an incorrect segmentation. When dealing with a font with a variable pitch, the other method is employed. In the variable pitch method, the distance between two letters is evaluated, and if that distance is more than a threshold based on the average letter separation, the algorithm adds a space. Figure 4 shows a correct segmentation of the characters in a variable pitch text.

Figure 4. Correct characters segmentation

The last step of OCR is to recognize each character individually. This can be done in different ways, but one popular approach is to use neural networks and other learning algorithms to convert the characters. The approach in the Tesseract algorithm is to use an adaptive classifier. It presents the advantage to be faster but is less likely to adapt to slight variations compared to neural networks.

After the OCR is completed, another extra step helps reconstitute the words and sentences. First, using the information about the spaces, the characters are grouped into words. Punctuation is also used to identify the beginning and end of each sentence. Once the sentences are identified, they are passed down to a Context-Free Grammar (CRG) algorithm, that decompose the sentence in its basic elements, such as verbs, nouns, adjectives, and so on. This process allows the detection and correction of spelling errors for each category of words separately using reduced portion of a word database. In case the algorithm can't detect a structure in a sentence, or if a single word is detected, then the algorithm reverts to a simple match with the word database.

One advantage that the proposed system has over a standalone device is the Internet access. When the image detects a text that can't be correctly recognized, as it could be the case for example with handwriting, the smartphone can use its Internet connectivity to send the image to a server for further processing. The servers have the ability not only to pack more processing power but also to have access to a greater database and therefore more complex recognition algorithms.

V. User Feedback

As mentioned earlier, the user feedback is divided into two categories. The localization, and the actual text extracted from the signs, as shown in the flowchart in Figure 5.

The localization feedback is linked to the signs detection, and provides insights as for where the signs are. When the user walks around a sign, it is unlikely that all the signs will be relevant to the situation. For example, a user going forward in a corridor doesn't necessarily need to read the signs that are posted on an announcement board. Or a user looking for a door among several in the field of view of the camera will need not only to read the signs, but know where the signs are to head towards the correct one. Therefore, the signs detection also locates the signs in the image, and forward the information to another algorithm in charge of formulating sentences to convert to speech and play to the user. The same algorithm can give instructions to turn or tilt if a sign is detected as being incomplete, or with a missing piece.

In comparison, reading the text to the user is a more trivial task as the text extracted from the image just needs to be played back to the user as is, after the CRG algorithm to make sure the text is somewhat understandable. However, the text cannot be played in any order. The text needs to be read in sequence with the information provided by the localization feedback algorithm in order to be clear to the user. For example, each text converted can be preceded by a message designating which sign is being read and in which location.

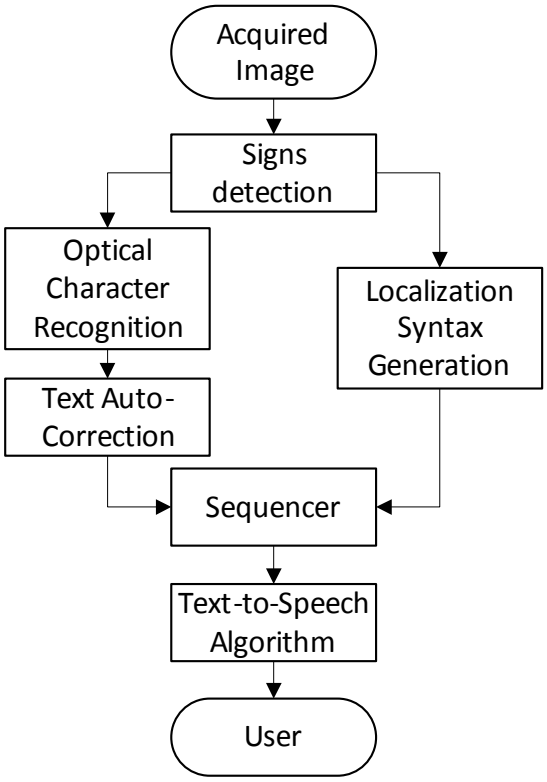

Figure 5. Flowchart of the system's feedback

VI. Experimental Results

In order to test the implementation, we used a Google Nexus 5X phone, running Android 6.0. The application was set to take still pictures, and then run the sign detection and OCR. The recognized text is then displayed over the image and by touching the sign on the screen, the application reads the text on the speaker. The samples used are some of the signs in the ECE department building at IIT.

Figure 6 shows the result of a successful capture and conversion of a door sign. In this case, both the laboratory name and the university logo were correctly recognized.

Figure 6. Original door sign (left) and as processed with the smartphone (right)

Figure 7 shows the algorithm processing a single sign with several components. We can see that the algorithm has issues associating all the different lines together, but otherwise manages successfully in recognizing the different words.

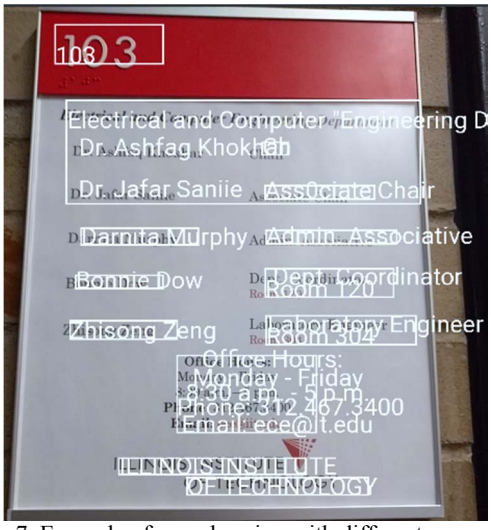

Figure 7. Example of complex sign with different components

In Figure 8, we can see the sign giving the direction for a group of doors, from 306 to 308. The limitation of the system however is that it doesn't detect the arrow and would not be truly helpful in the case of a user navigating to one of these rooms.

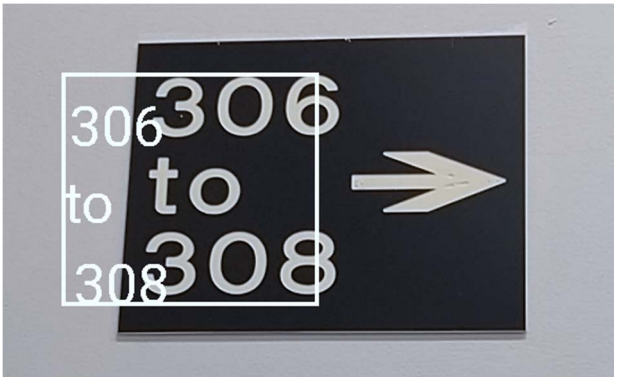

Figure 8. Sign with directions

Finally, Figure 9 shows a failed sign recognition. The second line is not detected, and the first line contains error that the error correction didn't manage to correct.

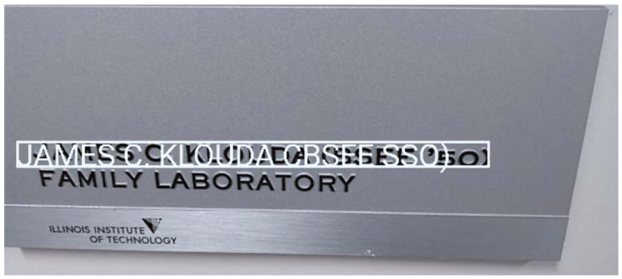

Figure 9. Door sign incorrectly recognized

VII. Conclusion

In this paper, we presented the design flow to implement a Text-to-Speech algorithm that can enable visually impaired people to "read" the signs in the environment. The experiments show the feasibility of the concept implemented on an Android smartphone, and using still images, which can be extended to a real-time implementation in the future.

References

[1] World Health Organization, "Visual impairment and blindness - Fact Sheet N°282," August 2014. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/. [Accessed January 2017].

[2] A. Nichols, "Why Use The Long White Cane?," 1995. [Online]. Available: https://web.archive.org/web/20100330050804/http://www.blind.net/g42w0001.htm.

[3] F. M. D'andrea, "A History of Instructional Methods in Uncontracted and Contracted Braille," *Journal of Visual Impairment & Blindness; New York,* vol. 103, no. 10, pp. 585-594, 2009.

[4] American Foundation for the Blind, "Technology Resources for People with Vision Loss," [Online]. Available: http://www.afb.org/info/living-with-vision-loss/using-technology/12. [Accessed January 2017].

[5] R. Smith, "An Overview of the Tesseract OCR Engine," *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007),* vol. 2, pp. 629-633, 2007.