Development of a prototype to facilitate the mobility of blind people through object recognition and text to speech technology

I. Sánchez López[1], R. García-Carmona[1], A. Otero[1]

[1] Escuela Politécnica Superior, Universidad San Pablo CEU, Madrid, España,
{ignaciosl96@gmail.com, r.garcia.carmona@gmail.com, abraham.otero@gmail.com}

Abstract

TagIt! is a prototype that enables blind people to get a better understand of their environment. It makes use of a camera, Wi-Fi connectivity and a speaker to capture images of the environment, process them looking for objects and describe such objects to the user using speech. It is presented in a wearable format that can be integrated in a standard white cane or worn.

1. Motivation

According to the World Health Organization (WHO), 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have poor vision [1]. This figure represents more than 3% of the actual worldwide population.

About 90% of the world's visually impaired people live in low-income settings, so they do not usually have access to modern solutions that could help treat or cope with their disability. They must perform their day to day tasks with the only help of a white cane.

The goal of this work is to provide a cheap solution that can improve the mental picture that blind people form of their environment. TagIt! does this by describing in plain language what objects are in front of it. This process is activated by a button that the user can push at will, after pointing at the direction the user wants to get information of. This way, blind people can expand the perception of their surroundings with just the press of a button.

Today´s technology allows us to analyze and describe pictures in a couple of seconds, using computer vision and machine learning. These techniques have been proved to be very effective and precise, so we thought that we could use them to help visually impaired people to have a wider feedback of their environment.

2. Characteristics and use of TagIt!

TagIt! has been designed to be attached to a white cane, close to the handle, by using a clamp. It has an external button that the user pushes when he or she wants to know which object the user is facing. The prototype has the following characteristics:

- Dimensions (H x W x D): 43x80x40 mm
- Weight with / without support: 136/116 g
- Micro-USB charging port
- A 5 Mpx camera, to take the photographs
- A speaker, to say out loud the name of the object
- A 2000 mAh battery

TagIt! is a very easy to use device: the user points the white cane to the object he or she wants to tag, presses the button and waits until TagIt! plays aloud the tag assigned to the object that the user is facing.

TagIt! has been designed to be attached to a white cane, so it includes an adapter that can be easily removed if the user wants to move the device, charge it or wear it in a different way. This adapter uses a simple ergonomic screw.

The button is held at the grip level, so it can be easily pressed when needed.

To give the voice a more natural feeling, the chosen TTS (text to speech) technology is Google´s Text-To-Speech [2], which recently updated the quality of its voice to deliver a more natural intonation, especially in the English language.

TagIt! connects to the Internet via Wi-Fi, so it can work under a pre-configured Wi-Fi access point or hotspot, which a standard smartphone can provide.

The device is actively waiting for the user input, but only takes photos when instructed. The estimated battery life is roughly 10 hours with an intense use. In idle mode, it can last up to 16 hours. When the battery runs out, the user must charge it using a generic micro-USB charger.

3. TagIt! Specifications & Design

The basic features we needed to build the prototype were: a microprocessor that is powerful enough but does not need too much power, a small but good quality camera and a speaker.

As the computing platform, we chose the Raspberry Pi Zero W [3]. This is an updated version of the well-known Pi Zero, that includes wireless connectivity, such as Wi-Fi and Bluetooth, integrated. The reasons for which we selected this microprocessor instead of other alternatives, such as an Arduino were:

- *Faster CPU:* enables a faster processing of the image and therefore gives the user a better experience.
- *Integrated modules:* unlike the Arduino Micro, the Pi Zero already includes Wi-Fi and Bluetooth, so we can save the space and time needed to work with external modules. The Wi-Fi module was especially important for this prototype, since it provides outwards connectivity.

- *Programming flexibility:* the Pi family of microprocessors are designed to run a light OS based on Linux. This allows us to use different programming languages if we need to. With the Arduino, we are forced to use the limited Arduino programming language.
- *Wider hardware solutions:* the Raspberry family is the most popular single board computers for do-it-yourself projects [4]. Over time, this has led to a massive amount of solutions, that can tackle specific problems.

The camera selected is a 5Mpx camera module for Raspberry Pi Zero from Pimoroni. Originally, a camera system called PIXY [5] was evaluated. This camera has a processing module with the ability to detect objects that have been previously stored in its memory. It uses a color pixel detection mechanism. However, since this camera is sold as a complete computer vision solution, it cannot be easily programmed and customized to perform functions other than those already designed for. The camera module included in PIXY was tried independently and, even when we managed to make it work in Arduino, it proved to be impractical, since data had to be stored in a memory card and another module would be needed to make this possible. Even with the microSD card, Arduino processing speed would have made the system too slow. Finally, we concluded that this camera provided no advantage over a smaller and cheaper camera that does not include the processing module.

In the end, we used a mini IR camera [6] specifically designed for the Raspberry Pi Zero. It has a considerably smaller size than the already presented alternatives. And, since the Raspberry already has a microSD port (the SD card contains the operating system), this media can also be used as the storage space for the images. We don´t really need to store the pictures for much time, since they can be overwritten once they are analyzed, but this feature could be used for debugging and troubleshooting purposes. This camera has also the advantage of having a higher resolution than PIXY. On top of that, it doesn´t have the infrared (IR) filter installed, so it can see a wider spectrum, which, even if it leads to some distortions of colors under specific conditions, provides better imaging at low light situations, like dusk, dawn or using streetlights as light source.

The speaker chosen is the Speaker pHAT [7] from Pimoroni. Phats are modules designed for Raspberry that lay over the main board as a “hat” (hence the name), using the GPIO rows both as mechanical support and a way of communication with the main board. Is a solution that allows us to build our system by simply putting the desired “hat” over the microprocessor. Hats can be exchange so the functionality provided can be easily customize.

Since we wanted the prototype to have a reasonable battery life, we added a 2000 mAh LiPo battery. This battery provides us with 10 hours of autonomy, which is more than enough for the desired use of the device. This battery has an output voltage of 3.7 V but the Raspberry Pi and the speaker need 5V, so we used a voltage converter, the Adafruit Powerboost 1000c [8]. Since it is designed to work with batteries, it also contains a charging circuit built-in with a micro-USB charging port.

Finally, we added a physical button, so the user can take a picture of the area he or she wants to know which objects contain . For this aim we used a general on-off switch.

3.1. Implementation

TagIt! has been designed with the Raspberry Pi Zero as its center. It has Raspbian - a Raspberry Pi version of Linux – installed in a microSD card. When powered, Raspbian is booted up and the OS starts running a Python script that runs the desired code, as explained in this section.

First all needed libraries are loaded. Immediately after this it configures the input pin for the button, which in our configuration corresponds to the pin 11. We needed to avoid accessing the pins that were already in use by the speaker, since the Phat modules use several of the GPIO pins, but not all of them.

Then there is an infinite loop that executes the main code that listens to the trigger generated when the button is pressed.

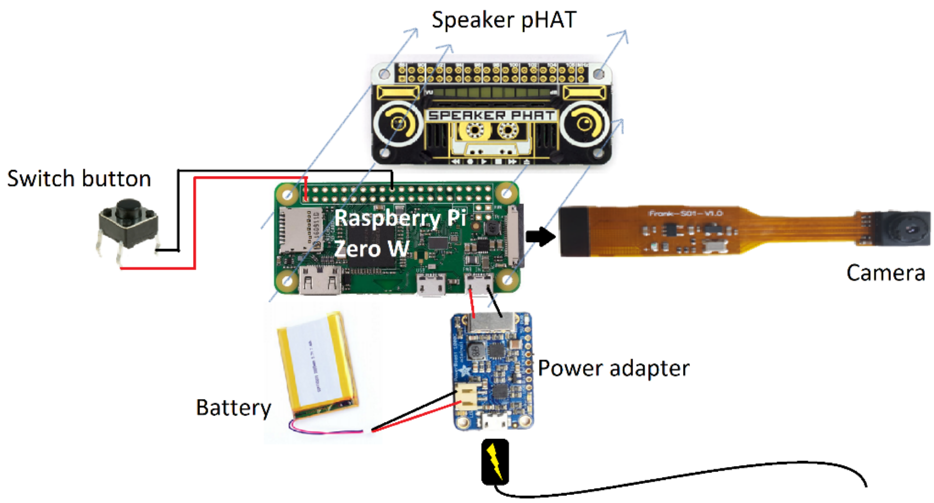

***Figure 1**. General scheme of TagIt!*

When this happens, the code calls the system function “raspistill”, which takes a photograph using the camera that the raspberry has connected. It´s strength lies in the wide array of configuration parameters, that allows us to easily modify the picture taken: size, quality, rotation, brightness, effects (such as negative), timer, etc. We use some of these to improve the image quality so the objects in it can be more easily recognized.

We put a timer of 1 second to this trigger, to make sure that the camera balances the brightness and contrast before it takes the picture, otherwise the image could be distorted, as the camera has to turn on and adapt to the environment before taken a photograph. We also specify the output of the image, as we want to save it and sent it to an Amazon Web Service (AWS) that provides computer vision on demand.

The service we chose, Rekognition [9], uses computer vision and machine learning to analyze the objects in an image and return the probabilities of different objects being there. We use this function to analyze our images and tell the user what is the most probable object, since that is probably the object the user is facing to.

To do so, we load the image to a bucket in AWS, which is the way documents are stored in the Amazon platform. Then, by calling the aforementioned Rekognition service, we analyze the image and retrieve a list of words with an associated probability, which represents the amount of confidence the system has on the fact that a given element is present in the picture.

For the sake of simplicity and to not overwhelm the user, we extract the name of the most probable element and read it out loud through the speaker. To transform the text into an audio file we use a TTS software designed to read text as similarly as possible to the way a human would do it. In this case we use the Google TTS, the same service Google employs in some popular applications like Google Maps or Google Now.

To use the Google TTS, we must send the object´s name to Google using a link. Then we retrieve from the remote server a sound file corresponding to the word we have just send. Using the mpg123 software [10], a very simple and easy to use tool to reproduce sound files, we are able to reproduce the file from the TTS through the speaker.

All this process takes approximately 6 seconds with a good connection; a reasonable timing for a device like this. Further improvements could reduce the time needed to complete a cycle significantly.

The code developed is available in GitHub [11].

3.2. External design

TagIt! Has been designed to be attached to a standard white cane (although it could be easily adapted to other positions). Therefore, the box that contains the prototype has been adapted to use a support installed in the cane. This will be attached using a screw that can be handled without additional tools and allows the user to control the tilting. The support can be easily attached and removed from the white cane, but at the same time it will stand firmly fixed.

The box has several openings: one for the camera, another to charge the device, and several smaller holes to let the sound pass through the wall to the exterior. Additionally, the strap is made of flexible rubber, for improved comfort.

Everything has been designed with 3D designing programs and has been built using a 3D printer. Figure 2 and 3 show the box design.

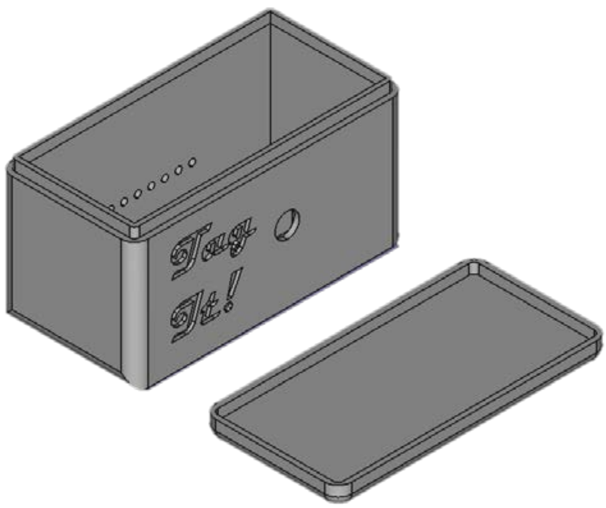

***Figure 2**. Virtual design of the box*

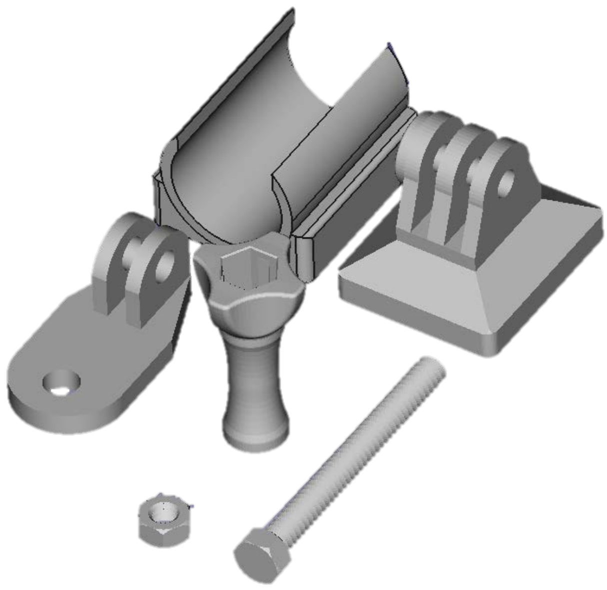

***Figure 3**. Virtual design of the support*

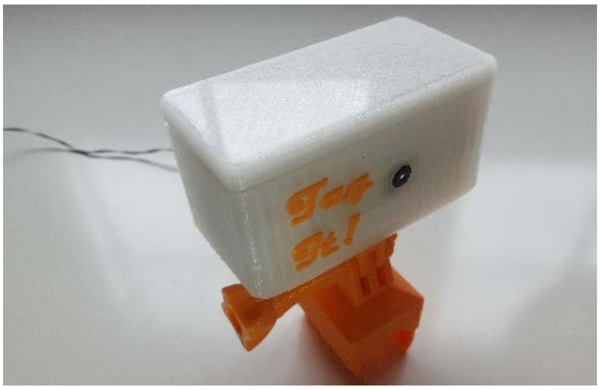

Figure 4. *Final design of TagIt!*

4. Conclusions

This prototype is a cheap and easy to use tool that attaches to a standard white cane and provides blind people with an improved awareness of their environment.

It is important to note that the user does not need to get rid of the white cane to use the device, something that blind people are not comfortable doing. We also believe that those who currently use a white cane will have no problem getting used to the device, thanks to its simplicity and easy handling. However, we would like to test the prototype with blind people, to better asses the truth of this claim. All in all, TagIt! has the potential to provide an important quality of life improvement to its user.

Acknowledgments

I would like to thank the professors at San Pablo CEU for their help and support in the realization of the project, as they have been crucial for the development of the prototype. Finally, I would like to thank Sarah Rance for her proofreading of this text.

References

[1] World Health Organization (WHO) web page. http://www.who.int/mediacentre/factsheets/fs282/en/ (Revised: October 2017).

[2] Wikipedia reference on Google´s Text-to-Speech. https://en.wikipedia.org/wiki/Google_Text-to-Speech (Revised: October 2017).

[3] Raspberry Pi Foundation web page, manufacturer of the Raspberry Pi Zero W https://www.raspberrypi.org/products/pi-zero-w/ (Revised: October 2017).

[4] Raspberry Pi Foundation explanation of what is a Raspberry Pi https://www.raspberrypi.org/help/faqs/#introWhatIs (Revised: October 2017).

[5] Charmed Labs web page, manufacturer of PIXY cam http://charmedlabs.com/default/pixy-cmucam5/ (Revised: October 2017).

[6] Pimoroni shop web page, designer and distributor of the camera https://shop.pimoroni.com/products/raspberry-pi-zero-camera-module (Revised: October 2017).

[7] Pimoroni shop web page, designer and distributor of the phat https://shop.pimoroni.com/products/speaker-phat (Revised: October 2017).

[8] Adafruit web page, manufacturer and distributor of the component https://learn.adafruit.com/adafruit-powerboost-1000c-load-share-usb-charge-boost/overview (Revised: October 2017).

[9] Amazon Web Services (AWS) web page https://aws.amazon.com/rekognition/ (Revised: October 2017).

[10] Mpg123 player web page https://www.mpg123.de/ (Revised: October 2017).

[11] GitHub link to the code. https://github.com/nachosl96/tagIt (Revised: October 2017).