Revista Brasileira de Computação Aplicada, November, 2019

DOI: 10.5335/rbca.v11i3.9455
Vol. 11, Nº 3, pp. 59–71
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ORIGINAL PAPER

INSIDe: Image recognition tool aimed at helping visually impaired people contextualize indoor environments

Elias Fank[1], Fernando Bevilacqua[1,2], Denio Duarte[1], Alesson Scapinello[3]

[1]Federal University of Fronteira Sul, Brazil, [2]University of Skövde, Sweden, [3]Federal University of Rio Grande do Sul, Brazil

*eliasfank@hotmail.com; fernando.bevilacqua@{his.se;uffs.edu.br};duarte@uffs.edu.br;asselhorst@inf.ufrgs.br

Received: 2019-05-16. **Revised:** 2019-08-14. **Accepted:** 2019-09-25.

Abstract

Visually impaired (VI) people face a set of challenges when trying to orient and contextualize themselves. Computer vision and mobile devices can be valuable tools to help them improve their quality of life. This work presents a tool based on computer vision and image recognition to assist VI people to better contextualize themselves indoors. The tool works as follows: user takes a picture ρ using a mobile application; ρ is sent to the server; ρ is compared to a database of previously taken pictures; server returns metadata of the database image that is most similar to ρ; finally the mobile application gives an audio feedback based on the received metadata. Similarity test among database images and ρ is based on the search of nearest neighbors in key points extracted from the images by SIFT descriptors. Three experiments are presented to support the feasibility of the tool. We believe our solution is a low cost, convenient approach that can leverage existing IT infrastructure, e.g. wireless networks, and does not require any physical adaptation in the environment where it will be used.

Keywords: Android system; computer vision; SIFT; Visually impaired

Resumo

Os portadores de deficiência visual enfrentam inúmeros obstáculos em seu processo de inclusão na sociedade. A visão computacional pode ser usada para uma maior qualidade de vida aos portadores de deficiência visual, contribuindo com a acessibilidade dos locais onde eles frequentam, além de auxiliar na resolução de dificuldades encontradas em seu cotidiano. Este trabalho apresenta uma ferramenta baseada em visão computacional e reconhecimento de imagem para assistir pessoas com deficiência visual na contextualização em ambientes fechados. A ferramenta funciona da seguinte maneira: o usuário tira uma foto ρ utilizando uma aplicação móvel; ρ é enviada para o servidor; ρ é comparada com imagens previamente cadastradas; o servidor retorna os metadados da foto mais parecida com ρ; finalmente, a aplicação móvel retorna o áudio para o usuário baseado nos metadados. O teste de similaridade entre as imagens do banco de dados e ρ é baseado na busca do vizinho mais próximos nos pontos chaves extraídos das imagens através de descritores SIFT. Três experimentos foram realizados para identificar a utilidade da ferramenta. Acredita-se que a solução proposta é de baixo custo e uma abordagem conveniente que pode utilizar a infraestrutura de TI existente, *e.g.*, redes sem fios, e não exige nenhuma adaptação física no ambiente onde a aplicação será utilizada.

Palavras-Chave: Android; deficiência visual; SIFT; visão computacional

1 Introduction

Visually impaired (VI) people face a set of challenges when trying to orient and contextualize themselves. Sighted people use an influx of visual information to obtain contextualization; however, VI people are prevented from using that information. Instead, they must rely on different sensorial cues, e.g., sound, touch,

and smell, to contextualize the surrounding elements and properly navigate the environment. Context information is essential for VI people to navigate both familiar and unfamiliar environments (Loomis et al., 1998, Bradley and Dunlop, 2005). Sight is also considered an important promoter of different activities, including motor, perceptive and mental ones, so visual impairment might reduce or limit the capacity of social inclusion (Bittencourt and Hoehne, 2006). Initiatives that aim to help VI people are essential to allow those individuals to be more independent, which improve their quality of life and participation in society. Two of those initiatives are tactile paving surfaces and Braille labels. Environments present different configuration and constraints that can prevent the use of such technologies, including the lack of space, the existence of obstacles, and the impossibility of applying changes to the place, e.g., historical site. Additionally, logistical and financial investments are required to properly and fully equip large environments. All those elements directly affect VI people, which compromise their navigation and contextualization capabilities.

Technological advancements in different fields, such as computer vision, introduced efficient and cost-effective solutions to help the VI community. Mobile devices equipped with cameras can be used to help VI people without the costs and logistics of installing tactile paving surfaces or Braille labels in existing environments, for instance. Mobile devices are particularly powerful because they can be used in places where the previously mentioned aiding methods could not be installed, e.g. old or historical sites. Solutions based on mobile devices and computer vision can provide accessibility for such places, requiring relatively simpler or no adaptations when compared to other solutions that require the installation of physical equipment (Helal et al., 2001). Software solutions are often based on the acquisition and processing of photos to provide aid, which is significantly less costly and demanding than actually applying physical changes to a given place. Mobile solutions have also been reported as a way of promoting social inclusion for VI people (Lima et al., 2017). Research conducted on that front have shown the use of wearable audio assistance and obstacle detection based on lasers, shoes and smart glasses (Walimbe et al., 2017). It is clear that a computational system running on hardware at a reasonable cost, e.g., smartphone, can allow VI users to navigate better and contextualize themselves. This initiative is an essential step towards social inclusion of VI people. In that light, we propose a tool named *Indoor Navigation for viSually ImpaireD* (INSIDe), which aims to help VI people to contextualize themselves in indoor environments. It uses computer vision for the recognition of images containing meta information that were previously added during a mapping phase. Users, who can be partially or wholly visually impaired, use a smartphone to analyze the environment, receiving audio feedback regarding objects ahead, e.g., doors, news panel, among others. The tool has been designed to work indoors, making use of existing IT infrastructure available in the place, e.g., wireless network. Those characteristics allow our solution to be low cost and easily integrated into a wide variety of places, including those unable to be physically adapted to follow accessibility guidelines.

Differently from previous work, INSIDe does not require marker tags to be placed in the environment, e.g., QR codes, to provide feedback to users. The only requirements regarding hardware are a smartphone, a server, and a wireless network to connect them. Recognition of objects is performed on the server using feature extraction, i.e., Scale Invariant Feature Transform (SIFT) descriptors (Lowe, 1999), to check for similarity between images. As a consequence, high processing power is not needed on the smartphone used by the VI person. We believe INSIDe is a convenient, low cost and easy to use solution that does not require physical changes to be applied to the environment where it will be used. We highlight the following contribution of the present work: (i) an empirical evaluation of the use of SIFT descriptors in the context of image detection aimed for contextualization of indoor environments; (ii) our solution is non-obtrusive and does not require changes to the environment where it is deployed; and (iii) it uses ordinary, low cost hardware and is able to use already existing resources available in the place, e.g., wifi network, as well as being highly adaptable, since changes that eventually happen in the place can be easily updated in the database. Finally, the proposed tool and its architecture are not limited to VI users; it can also help sighted people to contextualize unfamiliar places or possibly enhance the visual experience with contextualized audio feedback in places like museums.

The rest of this paper is organized as follows: section 2 presents the related work, and the following section presents the INSIDe tool, detailing its architecture and modules. Section 4 presents the description and results obtained with three experiments that were conducted to validate our proposed tool. Sections 5 and 6 present a discussion and the limitations of our tool, respectively. Finally Section 7 presents a conclusion and future work.

2 Related work

Use of computer vision for object recognition with the aim of helping visually impaired people navigate is a recurrent research topic (Jafri et al., 2014). Chaccour and Badr (2015) propose a system to aid VI people to navigate indoor environments by using a network of cameras mounted in the ceiling. Images are sent to a remote processing unit, which uses computer vision to detect the user location. Navigation information is then sent and read aloud in a mobile application being used by the individual. Croce et al. (2014) also presents a solution based on mobile devices to inform a VI individual if the current navigation direction is correct. The ideal navigation path is marked on the floor using painted/printed tags. The mobile application informs the user about the navigation by vibrating, i.e., vibration intensifies when the user navigates away

from the marked tags on the floor. Elloumi et al. (2013) present a similar approach which provides indoor navigation based on a pre-defined and marked route. During a learning phase, images of the environment captured at specific points of a pre-defined route are added to a database. During the usage phase, the images captured by the user's smartphone (suspended in the user's chest) are compared in real-time to the previously added images in the database. The matched database image is used to calculate the current angular deviation of the user related to the pre-defined route. That information is provided to the user to aid the navigation. The guidance the user receives is not based on the objects in the surroundings, but on the angles and the deviation from the pre-defined route.

Solutions based on custom hardware mounted on VI users are also found in the literature. Limna et al. (2009) present a stereo vision system that uses computer vision to provide users with information regarding the distance of objects. Two cameras mounted on the user's shoulders provide a video feed that is processed to detect objects and their position to the user. A sound alert is emitted when a possible collision is imminent. A downside mentioned by the authors is the high computational cost of the solution, which requires parallel processing to be feasible. Similarly Wenqin et al. (2011) propose a tool that provides users with audio feedback regarding information about planar surfaces and obstacles ahead. Two cameras are mounted on each of the user's shoulders, whose images are analyzed by computer vision techniques to find relevant information to the user. Tian et al. (2010) use a single, small camera attached to a hat or pair of glasses to provide users with information about doors. The procedure to detect doors uses a geometric door model that contains only lines and corners. Consequentially the method does not rely on appearance features of the doors, e.g., color and texture, but on its shape instead.

Other approaches found in the literature focus on using computer vision or human help to extract textual information from the environment. Ezaki et al. (2004) propose the extraction of text found in scenes to help VI people. The process relies on image segmentation and letter recognition based on two classifiers, namely Naive Bayes and Support Vector Machine (SVM). Bigham et al. (2010) propose the VizWiz system, which allows VI users to recruit help from crowd-sourcing websites where humans quickly perform the recognition. Using the camera of a mobile device, the user takes a picture of an object, asks a question about it, and within a short time interval receives an audio answer. Dapper Vision (n.d.) also propose a solution that uses crowd-sourced help and the Google Glass. Users take a picture of an object and make a question about, which is then analyzed and replied by humans from websites like Twitter or Amazon Mechanical Turk Platform.

Solutions based on geographical information without image recognition are also reported in the literature. Helal et al. (2001) present an outdoors navigation system based on entirely automated analysis of geographical data. The VI user must use a portable computer, which continually receives information from surrounding sensors and systems, such as wireless networks, geographic information systems (GIS), and GPS data. Similarly, Loomis et al. (1998) present a navigation approach based on GPS and a database of objects and their locations. The system provides information regarding the surrounding elements based on the data available in the database and the current geographical position of the user. Authors highlight the challenges and costs of using a GPS-based system for navigation.

Finally, approaches based on object recognition try to overcome the limitations of previously mentioned works. Deb et al. (2013) proposes a low-cost solution that guides VI people to detour obstacle in outdoor environments. However, any further information about the objects is not given to the user. Patel et al. (2018) propose a similar solution based on machine learning techniques to detect obstacles which relies on a multisensor system composed of a infrared sensor, a web camera, a ultrasonic sensor, and a Raspberry pi. All those sensors, however, increase the equipment requirements for such a solution to work, demanding a more specialized setup associated with more expensive hardware. Bagwan and Sankpal (2015) propose a solution called VisualPal that does not require environment adaptations, being able to recognize colors, brightness, and objects. VisualPal uses an artificial neural network running on Android systems to recognize objects, however its performance is not discussed by the authors since processing time for object recognition is not reported. Lastly, Matusiak et al. (2013) also rely on object recognition to aid in VI people navigation. In such approach, SIFT descriptors are used to recognize objects, but it is not clear how the descriptors are compared since previous images must be stored for comparison.

Our solution differs from previous work in several aspects. Firstly, we do not require physical changes or adaptations to the environment where it will be used, differently from approaches that rely on ceiling cameras (Chaccour and Badr, 2015) or marks on the floor (Croce et al., 2014). Secondly, our solution uses the camera of an ordinary smartphone and a wifi network to operate. No special setup or hardware is required, such as additional mounted cameras (Limna et al., 2009, Wenqin et al., 2011), multisensors (Patel et al., 2018), antennas (Helal et al., 2001), fluxgate compass and batteries (Loomis et al., 1998), or crowd-sourced help from real people (Bigham et al., 2010). Finally, previous work focuses on the angular deviation of users concerning a pre-defined route (Elloumi et al., 2013), while our approach focuses on the identification of objects in the environment based on their unique features. As a consequence, our solution is still able to identify objects even if they are moved from one position to another in the environment. Similarly to other approaches based on object recognition (Deb et al., 2013, Bagwan and Sankpal, 2015, Matusiak et al., 2013), our solution uses SIFT descriptors, the already existing network infrastructure of an environment,

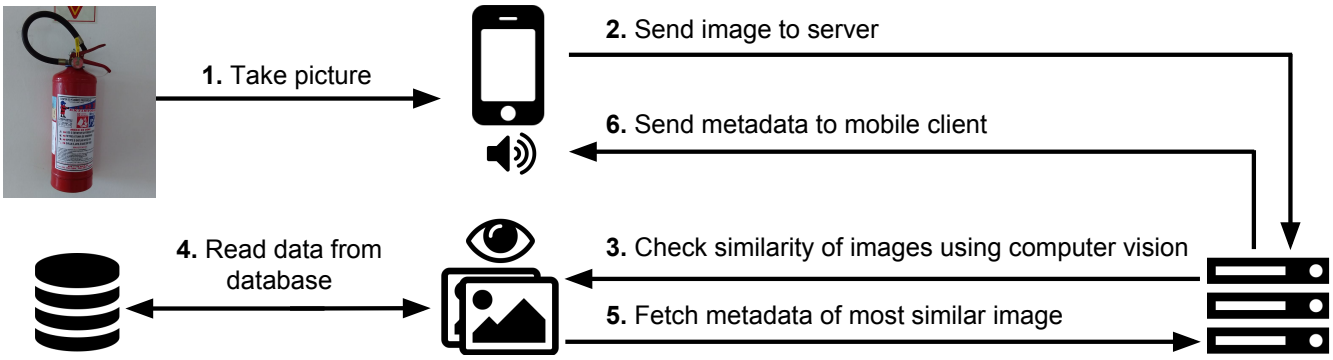

Figure 1: The overall architecture of our proposed tool. 1) Mobile client captures image; 2) Client image sent to server; 3 and 4) Check similarity of client image against images in database; 5) Fetch metadata of database image that is most similar to client image; 6) Send metadata back to mobile client, which will provide the user with an audio feedback

and a smartphone to provide a low-cost contextual guidance to visually impaired users. In that light, we clearly present the processing time and limitations regarding the use of SIFT descriptors in our approach, which is not clearly discussed by previous authors.

3 Tool overview

Our proposed tool, named INSIDe[1], aims at helping VI people to contextualize themselves better indoors. The process is composed of two main phases: environment mapping and object contextualization. In the mapping phase, the environment where the tool will be used has its objects mapped from pictures and metadata, e.g., the name of the object. All mapping data is uploaded to a web server to be accessed later by the mobile client in the contextualization phase. In the contextualization phase, shown in Fig. 1, a VI person uses a mobile application and the smartphone camera to contextualize him/herself with the environment. The contextualization process starts when the mobile application acquires a picture of the area in front of the user (step 1). The picture is sent to the server through the network (step 2), where computer vision techniques are used to search for a similar image stored in the database (steps 3 and 4). All images stored in the database were previously added during the mapping phase. When a similar image is found in the database, its associated metadata is retrieved (step 5), which includes a textual description of the image, e.g., door to room 20. Finally, the metadata is sent back to the mobile application, which reads the textual information aloud to the user (step 6). The contextualization process is repeated for each picture captured by the user, which allows him/her to obtain contextual information about the environment from the audio cues (based on the metadata returned by the server).

The following sections present in details the overall architecture and components of the mapping and the contextualization phase.

[1]Source code available at https://github.com/inside-project

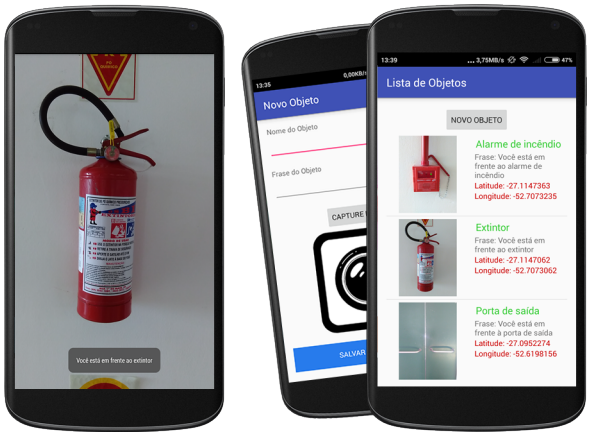

Figure 2: Mobile application INSIDe client. Left: application running in user mode. Right: application running in administrative mode.

3.1 Overall architecture

INSIDe uses a client-server architecture based on the HTTP protocol for communication. It allows the tool to leverage existing IT infrastructure and be easily deployed to environments with already working networks. The INSIDe client is a mobile application developed for Android 3.0 and above, featuring two working modes: user (default) and administrative, as illustrated in Fig. 2. When in user mode, the application contains a single screen that shows what is being captured by the device's camera. When the user taps any part of the screen, a picture is captured and sent to the server for analysis (see section 3.3). The administrative mode is intended to be used by a sighted individual who is responsible for providing the functionalities of the INSIDe tool in a given environment, including the process of updating any existing metadata when changes happen to the place. Finally, the INSIDe server has been

developed using PHP and the database management system MariaDB. The role of the server module is to provide mobile clients with contextual information upon receiving images and having them processed by the image recognition module (detailed in section 3.3).

3.2 Environment mapping

Environment mapping is a critical component of the tool. It is responsible for acquiring and curating all data required to allow our proposed solution to properly work, i.e., recognize objects and help VI users contextualize their navigation. The mapping process is based on two elements: the INSIDe client running in administrative mode, and the server module. The mapping begins with a sighted user, i.e., administrator, operating the mobile client in administrative mode while navigating the environment. The administrator must judge which objects are critical to the environment and whose contextualization will help VI users. For a given object of interest, the administrator maps it by taking a picture of the object and inputting information about it, namely: latitude, longitude[2], name, description, and name of the place where it is. Latitude and longitude are used to geo-reference the object, which narrows down the possible candidates during the search performed by the image recognition module. The name of the place is aimed exclusively to the administrator, so one can keep track of mapped objects and their locations when maintaining the database of images up to date.

When the server receives a new mapped entry, i.e., object picture and its metadata, the image is processed. Firstly, it is converted to grayscale and sized to have a maximum width or height of 500px (keeping the proportion among them). Next, a set of SIFT descriptors, i.e., key points, are extracted from the image, which are illustrated by the yellow circles in Fig. 3. Finally, the image, its SIFT descriptors, and metadata are stored in the database. The SIFT descriptors for any given mapped object are calculated just once, which optimizes the process of the image recognition module (section 3.3) when searching for similar images since stored descriptors can be used instead of recalculating them. The capacity to find similar images in the contextualization phase is directly related to the quality of mapping images. A mapping image is said to be of good quality when the administrator can adequately frame the mapped object while capturing the least amount of information surrounding such object, e.g., adjacent elements. If the mapping image of an object contains artifacts, e.g., a superposition with other objects, reflections, or occlusion, then extracted key points might not be unique enough to differentiate the mapped object. It causes wrong matches in the image recognition process, resulting in false-positive feedback to the user.

Ideally, the administrator should map the same object from different angles and distances. The search for images performed in the contextualization phase relies on the similarities of the images, so a picture taken by the mobile client is more likely to be found if it mimics the configuration (position and angle) of an image in the database. As an example, if the VI user is positioned to the left and 2m away from a mapped object, the recognition of the picture of that object is more likely to be found in the case the administrator captured a mapping image relatively close to the position where the VI user is, i.e., similar distance and angle.

3.3 Object recognition and contextualization

Object recognition is performed with the interaction among the mobile application, the server, and the image recognition module. The mobile application starts the process by capturing an image and sending it to the server over the network. The server receives the image and forwards it to the computer vision module, which performs the recognition. As previously explained, during the environment mapping phase several objects are pictured and added to the database along with their associated metadata. The recognition process is based on a similarity test performed on the image received from the mobile client and the object images stored in the database.

The similarity test among the image sent from the mobile client and the images stored in the database is based on the search of nearest neighbors in sets of key points extracted by SIFT descriptors. Given I_c as the image captured and sent to the server by the mobile client, and $M = \{I_0, I_1, ..., I_n\}$ as the set of images stored in the database, created during the environment mapping phase. Firstly the SIFT algorithm is applied to I_c, which extracts a set of key points, named F_c. Next, each image $I_i \in M$ has its SIFT-extracted key points retrieved, which build the set F_i. As previously mentioned, the key points of images in M are extracted via SIFT once when the image is added to the database. Following that step, a *kd-trees* matcher with five trees is used to calculate the similarity among the key points of F_c and F_i. We used the SIFT and *kd-trees* implementations provided by OpenCV and FLANN (Muja and Lowe, 2014) libraries, respectively. A coefficient p_i is calculated for each image I_i based on the percentage of matches found between F_i and F_c according to the FANN *kd-trees* comparison, as described by Eq. (1):

$$p_i = \frac{kdtrees(F_i, F_c)}{|F_i|} \tag{1}$$

where $kdtrees(a, b)$ represents the number of matches between the sets of key points a and b, and $|F_i|$ represents the cardinality of set F_i. As a result, p_i has a value within the range $[0, 1]$, where 0 indicates no match found, while 1 indicates a 100% match. After testing the similarity of I_c and the images in M, the image associated with the highest p_i value is selected

[2]Latitude and longitude can be automatically collected depending on the device used.

as the most similar one.

The INSIDe mobile client was designed to run on a wide range of devices. In some cases, however, the device's camera might capture images with a lower resolution compared to the images stored in the database. Consequentially, the image sent by the mobile client, i.e., I_c, produces fewer extracted key points, i.e., F_c, compared to the set of key points extracted from images stored in the database, i.e., F_i. In order to account for such problem and ensure the tool would work even with low-end smartphones, Eq. (1) was modified to account for the number of key points of I_c in the similarity test. Eq. (2) presents the adapted calculation:

$$p_i = \frac{kdtrees(F_i, F_c)}{|F_i| \times 0.7 + |F_c| \times 0.3} \quad (2)$$

In Eq. (2), matching key points are divided by a weighted mean derived from the number of key points available in both images being compared. For example, assuming I_i has 10000 key points, i.e., $|F_i| = 10000$, I_c (image to be compared) has 7000 key points, i.e., $|F_c| = 7000$, and $kdtrees(I_i, I_c) = 5500$, then the result of Eq. (1) would be 0.55, while the result of Eq. (2) would be 0.60. In our initial similarity tests with Eq. (1), a significant number of images that should be considered similar among each other produced low values for p_i due to small variations in the resolution of the images involved. Empirical tests have shown that Eq. (2) presented satisfying results for the similarity test compared to those of Eq. (1). Additionally, it better accounted for differences in resolution when the weight of I_i and I_c was 30% and 70%, respectively, producing fewer false-positives.

Fig. 3 shows a visual representation of the matching performed among the key points of I_i and I_c of two objects in the similarity test. On the left of Fig. 3 is the image sent by the mobile client, i.e., I_c, when performing a contextualization, which in this case is the identification of a fire extinguisher. On the right of Fig. 3 is an image stored in the database, i.e., I_i, which has been selected as the most similar, i.e., presented the highest number of matches among their key points and the key points extracted from the image sent by the mobile device. The blue lines highlight the matches among the key points of both images.

The similarity test has a significant computational cost, so the number of images to be tested in the search impact the overall time the process takes to complete. Consequently, the search for similar images has been optimized by geo-filtering the set of images M before performing any similarity test. As explained in section 3.2, images stored in the database during the environment mapping contain metadata with its physical location, i.e., latitude and longitude. The geographic position of the mobile client and the images in M is taken into account to reduce the number of images to be compared. Instead of testing the similarity of all images in M, only those whose Euclidean distance to the mobile client are less than D meters are considered. In the experiments presented in this paper, a value of 10m has been used for D.

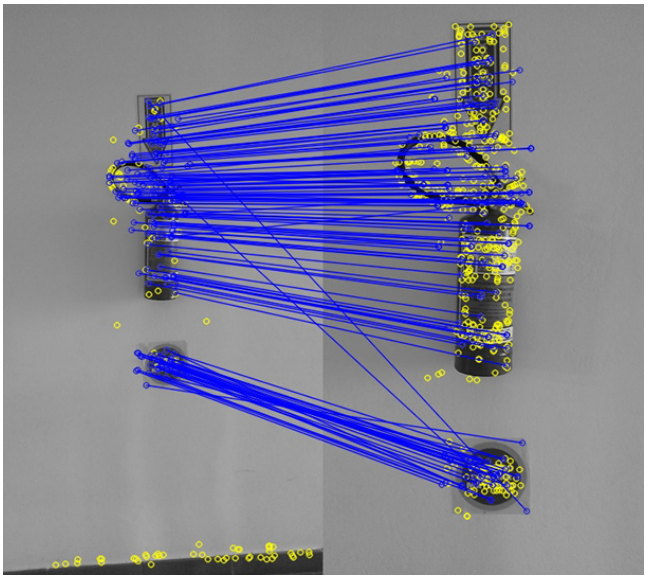

Figure 3: Comparison of key points between two images. Left: image sent by the mobile client featuring a fire extinguisher. Right: database image selected as the most similar to the mobile client image according to the matching of key points (blue lines).

4 Experimental validation

We conducted three experiments to validate the feasibility of our proposed tool. Experiments were designed to test different aspects of the tool, such as usability of the mobile application and the accuracy of objects recognition in an environment that has been mapped by the tool. All experiments were conducted at the dependencies of the Federal University of Fronteira Sul. In experiments 1 and 2, a female sighted student of the university consented to participate in the study after being informed of the experimental procedure. The subject was blindfolded to simulate the condition of a VI person, as illustrated in Fig. 4. In order to prevent the subject of using any previous knowledge about the environment and its objects, all objects included in the environment mapping phase of the tool received random names and descriptions, e.g., the fire extinguisher was mapped as the exit door, the news board was mapped as the elevator door, and so on. This procedure ensures the subject will not try to find any particular object by its real name and position but instead will use the names reported by the contextualization information provided by the INSIDe mobile application. Additionally, the researcher who conducted the experiments regularly monitored the movements of the subject, preventing any potentially unsafe collision against obstacles that were not accounted for, e.g., pillar or seat along the way. The following sections describe each one of the

Figure 4: Blindfolded subject using the mobile client to contextualize an object, i.e., fire extinguisher.

experiments, along with its objective, methodology, and achieved results. A discussion of the results is presented in Section 5.

4.1 Experiment 1

4.1.1 *Objective and methodology*

This experiment aims to evaluate if a subject can find a requested object in an environment of small proportions under controlled settings, i.e., following specific (and ideal) instructions regarding the use of the mobile application. In the context of our experiment, an environment of small proportions is a room with fewer objects, e.g., a classroom or a corridor. In the experiment, the subject must complete three tasks, i.e., T1 to T3. For each task, the subject was instructed to navigate the environment and locate a requested target object using the mobile application. The researcher experimenting randomly selected the target object from the pool of all mapped objects in that environment. The researcher also instructed the subject to take frontal pictures of the objects to be recognized/contextualized. The subject was also encouraged to rely on physical touch to locate potential objects to be contextualized by the mobile application. The subject was instructed to continue recognizing objects in the environment until the target object was found, which would conclude the task at hand. During each task, the following information was collected: distance between the target object and the subject when the task started, which objects were recognized during the contextualization of the environment until the task was completed, number of pictures sent by the mobile application to the server (which also corresponds to the number of contextualization actions performed by the user), and time the subject took to complete the task. Finally to evaluate the user experience regarding the mobile application, the subject answered with "Yes" or "No" to questions $Q1$ and $Q2$, which were framed as "Are you frustrated?" and "Regarding the feedback provided by the mobile application, did it help you complete this task?", respectively.

4.1.2 *Environment mapping*

The experiment was conducted in a long corridor at one of the university buildings. In total, 20 objects were mapped, i.e., added to the INSIDe database for recognition, namely: fire extinguisher, classroom doors (4 in total, whose room number ranged from 303 to 306), exit door, bathroom doors (4 in total, 2 were signaled special needs bathrooms), drinking fountain, fire hose container, lab doors (2 in total), manual call point for fire alarm activation, elevator doors, and a news board. All objects were mapped (following the procedure described in Section 3.2) with a single frontal picture, which tried to frame the object entirely with as few of its surroundings as possible.

4.1.3 *Results*

Table 1 presents the results of the experiment grouped by task. Columns denote the following: *T* is the number of the task, *Duration* is the time to complete the task, *Target (distance)* refers to the requested target object and its distance to the subject at the beginning of the task, *Pictures* is the number of pictures captured throughout

Table 1: Results of experiment 1 grouped by task (T)

T	Duration	Target (distance)	Pictures	Recognitions	Q1	Q2
1	6 min	Fire hose (15m)	5	lab door, fire extinguisher, manual call point for fire alarm activation, lab door	No	Yes
2	10 min	Elevator (20m)	9	lab door, manual call point for fire alarm activation, fire hose	Yes	Yes
3	5 min	Exit door (10m)	3	Elevator, news wall	No	Yes

the task, i.e., contextualization requests, *Recognitions* is the list of objects that were recognized while the subject navigated the environment until the target was found, and finally $Q1$ and $Q2$ present the answers provided by the subject for the questionnaire at the end of each task. The subject was able to successfully locate and contextualize the requested target object in all tasks, reporting that the received audio feedback was helpful. However, the subject reported being frustrated during task $T2$, even though the requested target object was found in such case. Such frustration could be attributed to the fact that task $T2$ requested a target object that was away from the subject (20m) and several contextualization requests were not successful during the navigation. During $T2$, the mobile application took nine pictures: five (55%) yield negative feedback, i.e. *object not mapped*, and four (45%) correctly recognized the objects being analyzed, among them the requested target object. It is important to highlight that the majority of the time used to complete the task was not associated with waiting to receive feedback from the mobile application. Instead, it was related to the subject navigating the environment carefully due to the blindfold condition.

4.2 Experiment 2

4.2.1 *Objective and methodology*

The objective of this experiment is to use the mobile application to recognize as many objects as possible in the environment. It aims to evaluate the feasibility and accuracy of the tool when providing the subject with contextualization about the environment in a use case that is closer to how the tool would be used outside an experimental setting. Differently from experiment 1, for this experiment, the subject was less constrained regarding how the pictures should be captured. The subject was instructed to take pictures 1 or 2 steps away from the objects; however, the orientation of the object was not required to be frontal to the camera, e.g., lateral pictures were allowed. Lateral pictures of objects, as opposed to perfectly frontal ones, are more likely to happen in a real use case of the mobile application, since a user groping an object will immediately try to contextualize it. The researcher instructed the subject that if the mobile application informed it was unable to recognize an object, the subject should try again after slightly adjusting for the new picture, e.g., change the angle of the camera or move to the right/left vaguely. If the mobile application did not recognize the object after three tries, the subject was instructed to ignore the object and continue with the contextualization of other elements.

4.2.2 *Environment mapping*

The experiment was conducted in a long corridor at one of the university buildings. In total, 20 objects were mapped, i.e., added to the INSIDe database for recognition, namely: fire extinguishers (2 in total), classroom doors (four in total, whose room number ranged from 303 to 306), exit doors (two in total), bathroom doors (four in total, two were signaled special needs bathrooms), drinking fountain, fire hose container (two in total), lab doors (two in total), manual call point for fire alarm activation, elevator doors, and a news board. All objects were mapped (following the procedure described in Section 3.2) with 16 pictures each, all acquired at different distances and angles concerning the object. Fig. 5 illustrates the positioning used by the administrator user when mapping objects of the environment. Mapping pictures were acquired by following half the circumference line of two circles of 1m and 2m of radius both centered at the object. In the inner circle, the object was photographed from 7 different positions, which were equally spaced among them. Similarly, in the outer circle, the object was photographed from 9 positions equally spaced among them. The blue block in the center of the figure represents the object being mapped, while the red marks are the positions where mapping pictures were taken. When the administrator performing the mapping was unable to stand on the desired positions, or when his/her view towards the object was occluded, e.g., interference caused by a pillar, the position was adjusted until the object could be properly framed and the offending obstacle could be ignored.

4.2.3 *Results*

Subject attempted to recognize and contextualize a total of 20 objects in the experiment. A total of 13 objects (65%) were correctly recognized by the mobile application, namely: two exit doors, two fire extinguishers, two toilets (those of the special needs), manual call point for fire alarm activation, news board, elevator door, two fire hose containers, and two lab doors. A total of six objects (30%) could not be recognized, even after the three allowed retries. Finally, one object (5%) was wrongly recognized (false-positive): a fire hose container was recognized as an exit door.

4.3 Experiment 3

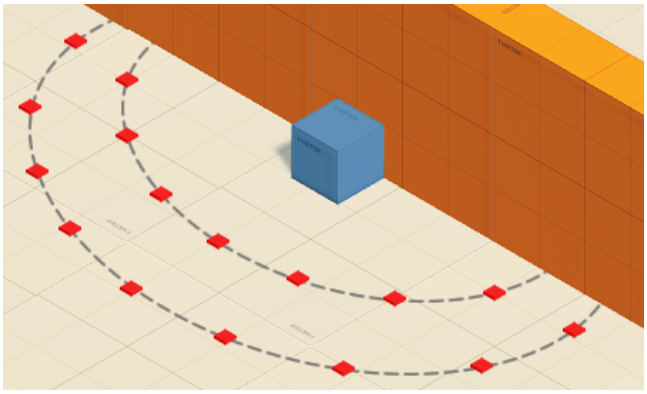

Figure 5: Visual representation of the process used to map a given object during experiment 2. The blue box represents the object, and the red marks are the positions from where pictures were taken to map that object.

4.3.1 *Objective and methodology*

The aim of experiment 3 is to evaluate how effective and robust the matching of SIFT descriptors is for the comparison of images of a given object pictured in different angles and distances. Differently, from previous experiments, the focus of this experiment is not on the user experience or on testing the full architecture of our solution. Instead, the focus is on evaluating the impact that different angles and distances have when matching key points. Consequentially, the user of the mobile client during the experiment was one of the authors, who was not blindfolded. Three objects were randomly selected to be used in the experiment: elevator door, fire extinguisher, and exit door. Each object was mapped using a single image, i.e., control image hereafter referred to Υ, which was taken in a frontal position at an ideal distance, i.e., enough to frame as much of the object as possible without capturing the object's surrounding elements/environment. Each of those Υ was then tested using nine images acquired from nine test cases. In test cases 1 to 3, each testing image was taken in front of the object (from the ideal distance), 1m away from such ideal distance, and 2m away from such ideal distance, respectively. In test cases 4 to 6, each testing image was taken left of the Υ position, from an ideal distance, then 1m and 2m away from it, respectively. Similarly in test cases 7 to 9, each testing image was taken right of the Υ position, also from the ideal distance, then 1m and 2m away from it, respectively. Each test case produced an image of the given object being tested, which was compared solely with the Υ of that particular object, i.e., no search was performed on the database. Images produced during the test cases of a given object contain variations in the angle and distance of the mobile client relative to the angle and distance used to capture such given object's control image. Test case 1 reproduces the exact setup of the control image, i.e., same angle and distance, and can be seen as the Υ position. In test cases 2 and 3, the user is in front of the object (same angle as Υ); however, the distance is different from Υ. In test cases 4, 5, and 6 the user is to the left of Υ position, aiming at the object (angle differs from Υ), and standing at various distances (including the same distance used in Υ). Similarly in test cases 7, 8, and 9 the user is to the right of Υ position, aiming at the object, and standing at various distances. When performing all test cases on Υ of each of the three selected objects, the user tried to frame the target object as best as possible. The experiment was focused on testing how robust the SIFT descriptors are at matching an object using images at different angles and distances from Υ. To ensure the analysis was indeed focused on the matching process, the mapping image, i.e., Υ and the ones captured by the mobile client during the test cases presented the same resolution, i.e., width and height. Additionally, we used Eq. (1) instead of Eq. (2) to calculate the matching between the key points of two images precisely because Eq. (1) assumes both images have the same resolution. The use of Eq. (1) should maximize the focus of the analysis on the matching of key points by eliminating steps used to account for different resolutions between the images being compared.

4.3.2 *Results*

Table 2 shows the percentage of matches found when comparing Υ of a given object against the images produced during all test cases performed on that object. For all three objects, test case 1 (same angle and distance as Υ) yield a 100% match among the key points of the images being compared. This result is expected since both images, i.e., control's and test case's, are the same. Excluding test case 9 for the elevator door, all other test cases presented a deterioration in the percentage of matches as the mobile client moved away from the target object. For the fire extinguisher and the exit door, the percentage of matches obtained when standing to the left or the right of the control position (test cases 4 and 7 respectively), at the same distance of Υ, were similar among each other. The elevator door, however, presented significantly lower matching percentage for test case 4 compared to test case 7. It could be explained by different lighting conditions or reflections in the target object when faced from another angle, since both test cases 4 and 7 present the same distance to the target. Overall all objects have shown a higher percentage of matches for test cases 2 (frontal, 1m away from control) compared to test cases 5 (left, 1m away from Υ) and 8 (right, 1m away from Υ). It supports the idea that the mobile client can acquire images of objects with different angles and distances relative to the one mapped in the database and still produce matches within a specific range.

5 Discussion

Results obtained from the conducted experiments show the feasibility of our proposed INSIDe tool. Even though the tests were conducted in a significantly small scale and without visually impaired subjects, our empirical results suggest that the image recognition based on

Table 2: Percentage of matches obtained in the comparison between control and test case images of different objects in experiment 3

	Object		
Test case	Elevator door	Fire extinguisher	Exit door
1	100%	100%	100%
2	38,65%	26,61%	42,45%
3	26,97%	17,59%	26,26%
4	15,67%	23,31%	28,72%
5	8,74%	5,63%	11,08%
6	7,54%	5,15%	8,64%
7	41,92%	20,67%	28,81%
8	31,28%	9,31%	22,84%
9	37,83%	6,22%	16,72%

feature detection aimed at object contextualization indoors is plausible. The proposed architecture used by the tool has been proven functional, allowing environments to be easily mapped without the need of physically adapting the place, e.g., the addition of QR code tags. However, the components of our solution are significantly affected by different elements. One of them is the quality of the images used during the environment mapping phase, which is directly related to how the mobile client captures images. As demonstrated by the experiments, ideal images to map an object, i.e., frame object as best as possible without its surroundings, does not necessarily yield an accurate object recognition. During the use of the mobile client by a VI person, captured images can negatively impact our proposed recognition algorithm, i.e., similarity test of images, if the object being contextualized is not framed correctly. The image framing is a significant limitation of our approach, particularly if the mobile client is capturing images where the object is partially cropped or the angle and distance of the picture considerably differ from the one used in the mapping process. This limitation, however, can be mitigated during the environment mapping phase by ensuring that a given object being mapped has several images taken from different angles and distances. Observations and results obtained during experiments 2 and 3 highlight such limitation along with possible improvements achieved when trying to mitigate the problem. In most of the cases during the experiments, the user successfully received audio feedback regarding the object being contextualized. In other cases, however, the mobile client reported that the object could not be recognized. Such negative feedback also happened when the subject was in front of the object, standing at an ideal position after groping the target, which is notably frustrating user experience. As mentioned, limitations regarding object contextualization can be mitigated; however, a definitive solution is a considerably complicated matter. Several factors affect the recognition procedure, such as the wrong orientation of the mobile device, which is a challenging problem to be solved via software.

Environmental conditions, e.g., different illumination, also affect our solution. As demonstrated by experiment 2, which was designed to simulate a real use case of the tool, the mobile client was unable to recognize some objects, even after the three allowed retries. Out of the 20 objects that were evaluated, seven were not recognized. Objects affected by different lighting conditions or that are too similar to other objects, e.g., doors to rooms and bathrooms, affect the recognition procedure. In some extreme cases, the recognition process can be affected to the extent that the key points extracted from the target image are not unique enough, which leads to false-positive results, i.e., mobile client wrongly recognizes an object. Experiment 2 presented those extreme conditions when the subject requested a contextualization while in front of a fire hose container, however, the mobile client reported the object as being an exit door. Any system aimed at helping VI people to navigate or contextualize themselves should not have false-positives since those might put the user at risk. According to the results of experiment 2, our tool presented only a single false-positive audio feedback. All other objects were correctly recognized or, in the worst case, the reported audio feedback informed the object could not be contextualized. The ratio of false-positive detections in our solution is significantly affected by the quality and amount of mapping images of each object (see Section 4). If an operator mapping a particular object in a given location acquires pictures of such object from several different angles, possibly mimicking the images that a user would take of such object, then that information is more likely to allow the tool to recognize the object in the future correctly. This claim is supported by the results of experiment 3, which have shown a better percentage of matching key points between images featuring an object pictured in similar distance and angle. Even though a higher number of mapping images per object in varying angles and distances might increase the accuracy detection, it negatively impacts the overall performance of the system, i.e., the time the tool takes to give audio feedback to users after they take a picture. As described in Section 3, the tool compares the matching points of an image taken by the mobile client against the matching points of several images stored in the database. Even though geolocation information is used to limit the number of images to compare, if an object stored in the database has several images associated with it, more comparisons will be performed in the search.

In order to investigate the performance impact of comparing database images, we conducted an empirical test focused on the time to perform comparisons concerning the number of images stored in the database. We stored in the database N images of random objects captured in various angles and distances, all with the same resolution, i.e., width and height. One of those N images was randomly selected and compared against such set of N images, using the comparison procedure of matching key points as described in Section 3.3. To account for any possible internal optimizations performed by the database and the operational system, e.g., page swap and different workload, each image is compared to all other N images 25 times ($N \times 25$ comparisons in total). The average time

Table 3: Mean comparison time, in seconds, of a given image against a set of N stored images

N	Time (no cache)	Time (cached)
20	000.41 ± 0.005	00.12 ± 0.0002
40	000.81 ± 0.022	00.24 ± 0.0018
80	001.64 ± 0.030	00.59 ± 0.0026
160	003.35 ± 0.034	01.36 ± 0.0507
320	006.66 ± 0.062	02.72 ± 0.1580
640	013.34 ± 0.155	05.39 ± 0.1583
1280	026.62 ± 0.232	10.76 ± 0.2493
2560	053.26 ± 1.144	21.51 ± 0.5040
5120	106.63 ± 3.720	43.08 ± 1.2200
10240	213.38 ± 9.481	86.08 ± 2.1457

among all those 25 repetitions is reported as the time it takes to compare a single image against a given set of N images in the database. Additionally, we investigated the impact of removing our cached key points, which is an optimization step used to prevent the recalculation of key points of any tracked objects/images stored in the database. When the caching of key points is disabled, SIFT key points must be recalculated for any comparison among the mobile client image and the images stored in the database. Table 3 presents the results of such test, which was performed on a single machine running Ubuntu Linux 16.04 (64 bits) with 8GB of RAM, Intel Core i5 processor (3470 @ 3.20 GHz) and a disk of 1TB (7200 RPM). As observed in both columns, *Time (no cache)*, i.e., caching of key points is disabled, and *Time (cached)*, i.e., caching of key points is enabled, the time to compare an image against a set of stored images significantly increases relative to N. The increase in time is linear and proportional to N. It is also possible to observe that caching the calculation of key points of images stored in the database drastically impacts the performance and response time. Caching key points of stored images in the database is essential to reduce the search time, which helps to deliver audio feedback to users as quickly as possible.

In light of our results, we believe that our proposed solution can be improved with further refinement of the image recognition procedures and adequate guidelines to be followed during the environment mapping phase. Our solution has low installation and maintenance cost and does not require physical changes to the environment where it will be used. It is a valuable initiative to increase the independence level of VI people in a variety of places, thus enhancing their quality of life.

6 Limitations

Some limitations of the experimental procedure and our tool should be noted. Firstly, our experiments had a significantly small sample size ($N = 1$) who was not a visually impaired person. It limits the extent to which our tool could be evaluated, so derived conclusions cannot be generalized. However, the experiments validated the feasibility of the tool, particularly regarding the proposed architecture and the validity of using SIFT descriptors, i.e., key points, to check the similarities between images. Secondly, it could be argued that our experimental design does not reflect a proper use case of a tool to possibly help visually impaired people because our subject groped for objects. A groping action could lead to the immediate identification of an object, bypassing the need of using a mobile application for that matter. As previously explained, we mitigated that problem by assigning random labels to objects in the experiment. Consequentially, a fire extinguisher could have been labeled as "exit door", for instance, so groping it still required the subject to use the mobile client to identify the object within the context of our experiment correctly. It is our understanding that visually impaired people indeed use groping and other instruments, e.g., probing cane, to contextualize themselves with the environment and objects through physical contact. Objects that are identical to the groping touch, e.g., doors without Braille labels, however, do require additional aid to be appropriately identified and contextualized. Those are the cases we believe our tool could be used. Another possible limitation of our tool is the use of SIFT descriptors instead of machine learning for the identification of objects. We use FANN kd-trees algorithm to calculate the similarity between the descriptors of two images (see Eq. (2)), using a matching threshold to evaluate similarity. Consequentially, our tool does not rely on any machine learning algorithm to recognize or match images (see (Wan et al., 2014, Turaga et al., 2008, Liu et al., 2017) for more details). Machine learning techniques are more likely to identify images accurately and robustly under challenging circumstances, i.e., different illumination. Although such techniques are powerful for computer vision and related fields, we believe that our straightforward approach is more suitable for a low-cost solution aiming to help visually impaired people contextualize their navigation. Our tool requires a mapping step that is easy to perform, uses an ordinary smartphone, and can rely on the existing infrastructure available in the environment, e.g., wifi network. Additionally, it does not require any model training time. Besides, our experiments have shown that image descriptors and similarity algorithms are efficient techniques for matching images for the contextualization and aid of visually impaired users.

7 Conclusion

This paper presented a tool aimed at helping visually impaired people to contextualize themselves during the navigation of indoor environments. The proposed solution is based on mapping the environment by adding pictures of objects of interest, e.g., doors and news boards, and their associated metadata, e.g., object's name, to a database. Using an application running on a mobile device, the visually impaired user takes a picture of an object/place whose information is desired. The image is then sent over the network to a server, which uses computer vision techniques,

i.e., feature detection using SIFT descriptors, to search for an image previously added to the database during the mapping phase that is similar to the one taken by the user. The database image whose similarity with the image sent by the user is highest is selected, and its metadata is returned to the mobile client. Finally, the mobile client reads the metadata aloud, i.e., audio feedback with the name of the object in the picture. Our proposed tool has been validated with three experiments. The first one aimed at evaluating the accuracy of the tool when recognizing objects in a small environment with ideal usage conditions. The second experiment focused on simulating the use of the tool that is closer to a real use case, whose conditions are more challenging. Finally, the third experiment explored how robust and flexible is the use of SIFT descriptors when checking for similarity between images. Results of the experiments show the feasibility of our tool. Further research is required to understand the limitations and accuracy of the proposed approach better; however, our empirical analysis suggests the tool is a plausible and low-cost solution to help visually impaired people. Differently, from other tools aimed at helping visually impaired individuals, e.g., tactile surfaces or Braille signs, our solution does not require any physical change to the place where it will be used. Additionally existing IT infrastructure available in the place, e.g., wireless network, can be leveraged by the tool, which further reduces deployment costs. It is our understanding that our proposed tool can be adapted to other use cases than helping visually impaired individuals. For example, sighted users might also use the tool to contextualize themselves in a given environment, such a museum or a sightseeing visit to an archaeological site. Future work includes further validation of the tool by using it on a larger scale than the one presented in this paper. Contextualization and navigation of a complete building, for instance, could be explored. Additionally, the environment mapping phase could be improved with the use of mapping guidelines. As presented and discussed in this paper, the quality of the images used in the mapping phase is an essential aspect of our proposed tool. Further investigation of such aspect could improve the overall accuracy of the tool, making it more robust and less likely to produce false-positives. Finally, field tests with real visually impaired subjects are the next step to accurately measure and improve the user experience of our tool.

Acknowledgements

This work was supported by FAPESC (Fundação de Amparo a Pesquisa e Inovação do Estado de Santa Catarina) - Edital 01/2014/FAPESC/Universal and Elias Fank was partially supported by Federal University of Fronteira Sul - EDITAL Nº 294/UFFS/2015 PRO-ICT/UFFS.

References

Bagwan, S. M. R. and Sankpal, L. (2015). Visualpal: A mobile app for object recognition for the visually impaired, *2015 International Conference on Computer, Communication and Control (IC4)*, IEEE, pp. 1–6. http://dx.doi.org/10.1109/IC4.2015.7375665.

Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S. et al. (2010). Vizwiz: nearly real-time answers to visual questions, *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ACM, pp. 333–342. http://dx.doi.org/10.1145/1866029.1866080.

Bittencourt, Z. Z. L. C. and Hoehne, E. L. (2006). Qualidade de vida de deficientes visuais, *Medicina (Ribeirao Preto. Online)* **39**(2): 260–264. https://doi.org/10.11606/issn.2176-7262.v39i2p260-264.

Bradley, N. A. and Dunlop, M. D. (2005). An experimental investigation into wayfinding directions for visually impaired people, *Personal and Ubiquitous Computing* **9**(6): 395–403. https://doi.org/10.1007/s00779-005-0350-y.

Chaccour, K. and Badr, G. (2015). Novel indoor navigation system for visually impaired and blind people, *Applied Research in Computer Science and Engineering (ICAR), 2015 International Conference on*, IEEE, pp. 1–5. http://dx.doi.org/10.1109/ARCSE.2015.7338143.

Croce, D., Gallo, P., Garlisi, D., Giarré, L., Mangione, S. and Tinnirello, I. (2014). Arianna: A smartphone-based navigation system with human in the loop, *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, IEEE, pp. 8–13. http://dx.doi.org/10.1109/MED.2014.6961318.

Dapper Vision, I. (n.d.). Open shades. Available at http://www.openshades.com.

Deb, S., Reddy, S. T., Baidya, U., Sarkar, A. K. and Renu, P. (2013). A novel approach of assisting the visually impaired to navigate path and avoiding obstacle-collisions, *2013 3rd IEEE International Advance Computing Conference (IACC)*, IEEE, pp. 1127–1130. https://doi.org/10.1109/IAdCC.2013.6514385.

Elloumi, W., Guissous, K., Chetouani, A., Canals, R., Leconge, R., Emile, B. and Treuillet, S. (2013). Indoor navigation assistance with a smartphone camera based on vanishing points, *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, IEEE, pp. 1–9. https://doi.org/10.1109/IPIN.2013.6817911.

Ezaki, N., Bulacu, M. and Schomaker, L. (2004). Text detection from natural scene images: towards a system for visually impaired persons, *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 2, IEEE, pp. 683–686. https://doi.org/10.1109/ICPR.2004.1334351.

Helal, A. S., Moore, S. E. and Ramachandran, B. (2001). Drishti: An integrated navigation system for visually impaired and disabled, *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, ISWC '01, IEEE Computer Society. https://doi.org/10.1109/ISWC.2001.962119.

Jafri, R., Ali, S. A., Arabnia, H. R. and Fatima, S. (2014). Computer vision-based object recognition for the visually impaired in an indoors environment: a survey, *The Visual Computer* **30**(11): 1197–1222. https://doi.org/10.1007/s00371-013-0886-1.

Lima, A., Mendes, D. and Paiva, S. (2017). Mobile solutions for visually impaired people: Case study in viana do castelo historical center, *Information Systems and Technologies (CISTI), 2017 12th Iberian Conference on*, IEEE, pp. 1–6. https://doi.org/10.23919/CISTI.2017.7975993.

Limna, T., Tandayya, P. and Suvanvorn, N. (2009). Low-cost stereo vision system for supporting the visually impaired's walk, *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology*, ACM, p. 4. https://doi.org/10.1145/1592700.1592705.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications, *Neurocomputing* **234**: 11–26. https://doi.org/10.1016/j.neucom.2016.12.038.

Loomis, J. M., Golledge, R. G. and Klatzky, R. L. (1998). Navigation system for the blind: auditory display modes and guidance, *Presence: Teleoperators and Virtual Environments* **7**(2): 193–203. https://doi.org/10.1162/105474698565677.

Lowe, D. G. (1999). Object recognition from local scale-invariant features, *The proceedings of the seventh IEEE international conference on Computer vision*, IEEE. https://doi.org/10.1109/ICCV.1999.790410.

Matusiak, K., Skulimowski, P. and Strumiłło, P. (2013). Object recognition in a mobile phone application for visually impaired users, *2013 6th International Conference on Human System Interactions (HSI)*, IEEE, pp. 479–484. https://doi.org/10.1109/HSI.2013.6577868.

Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **36**. https://doi.org/10.1109/TPAMI.2014.2321376.

Patel, C. T., Mistry, V. J., Desai, L. S. and Meghrajani, Y. K. (2018). Multisensor-based object detection in indoor environment for visually impaired people, *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 1–4. https://doi.org/10.1109/ICCONS.2018.8663016.

Tian, Y., Yang, X. and Arditi, A. (2010). Computer vision-based door detection for accessibility of unfamiliar environments to blind persons, *International Conference on Computers for Handicapped Persons*, Springer, pp. 263–270. https://doi.org/10.1007/978-3-642-14100-3_39.

Turaga, P., Chellappa, R., Subrahmanian, V. S. and Udrea, O. (2008). Machine recognition of human activities: A survey, *IEEE Transactions on Circuits and Systems for Video technology* **18**(11): 1473. https://doi.org/10.1109/TCSVT.2008.2005594.

Walimbe, A. A., Rao, S. S., Sureban, A. K. and Shah, M. S. (2017). Survey on obstacle detection and its notification through an android app for visually impaired people, *ASIAN JOURNAL FOR CONVERGENCE IN TECHNOLOGY (AJCT)-UGC LISTED* **3**. https://doi.org/10.1212/ajct.v3i3.244.

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y. and Li, J. (2014). Deep learning for content-based image retrieval: A comprehensive study, *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, pp. 157–166. https://doi.org/10.1145/2647868.2654948.

Wenqin, S., Wei, J. and Jian, C. (2011). A machine vision based navigation system for the blind, *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, Vol. 3, IEEE, pp. 81–85. https://doi.org/10.1109/CSAE.2011.5952638.