# Support model for navigation on sidewalks for visually impaired persons

Marcelo Cabral Ghilardi
PUCRS, Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação
Porto Alegre, RS, Brazil
marcelo.ghilardi@acad.pucrs.br

Isabel Harb Manssour
PUCRS, Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação
Porto Alegre, RS, Brazil
isabel.manssour@pucrs.br

***Abstract*—There are more than 35.7 million people with visual impairment in Brazil, of whom 6.5 million are totally blind or have great difficulty to see, even with the use of lenses. Although there are studies related to navigation support in urban environments for individuals with visual impairment, there have still been gaps to be researched. The main goal of this work[1] is to present a model that, based on computer vision techniques, can aid these people to walk on sidewalks. This model offers an integrated solution for localization and identification of tactile paving surfaces, detection of aerial and ground obstacles, and localization of crosswalks. Experimental results are presented and demonstrate the viability of this approach.**

***Keywords*-Computer Vision; Image Processing; Accessibility; Obstacle Detection; Visually Impaired; Tactile Paving.**

## I. Introduction

According to data from the World Health Organization[2], there are approximately 285 million visually impaired people worldwide. Among them, 246 million have low vision (less than 30%), and 39 million are blind. Blind tracks, white canes and guide dogs are typically used to help visually impaired people to walk outside [1]. However, even with these resources, mobility autonomy in outdoor environments presents a major challenge for them.

Nowadays mobile devices have shown a huge potential in supporting people with disabilities[2]. In addition, a wide number of assistive technologies have been proposed to support people in everyday activities. Several of these solutions are being developed based on images acquired by smartphones and GPS information (embedded in many modern smartphones). However, there are some challenges associated with the use of such images, such as the need to correctly point the camera to the target subject, image instability and real time processing. In these solutions, resources as vibration and sound of the device are used to provide feedback to the user.

Considering this context, the main goal of this work is to present a model, based on computer vision techniques, for helping people with visual impairment to walk on sidewalks. This model offers an integrated solution for localization and identification of tactile paving surfaces, detection of aerial and ground obstacles, and detection and localization of crosswalks. In order to prevent any misunderstandings, crosswalk localization is here considered as the relation between the detected crosswalks to the user, whilst crosswalk detection is related to its identification/recognition in image coordinates.

The main contributions of the proposed model are: (1) Identification of directional and warning paving surface in three different colors with an accuracy of 88.48%, and the possibility to achieve real time processing; (2) Detection of crosswalks with high accuracy rates (about 96.9% of accuracy), requiring low computation time (about 497 ms per image); (3) Localization of crosswalks with minor user intervention, with high accuracy rates (about 92.7% of accuracy); (4) Possibility to detect obstacles on the sidewalk by analyzing only the color patterns.

This paper is structured as follows: In Section II we provide an overview of some related approaches. The proposed model and implementation are detailed in Section III. In Section IV we present some experimental results. Finally, our conclusions and suggestions for future work are presented in Section V.

## II. Related work

There are different solutions to detect crosswalks: some of them require the user to take photos of the environment [1]; others need very high resolution aerial images [3] or are based on buzzers to indicate go and stop status [4] in traffic lights. Other commercial apps can provide spatial guidance for the user, but none of them propose to detect and locate crosswalks near the user. In our model, low resolution satellite and road map images are combined with GPS coordinates, without depending on buzzers or photos taken by the user. Moreover, it could be easily adapted to different crosswalk standards.

The shape and color of tactile paving surface do not have a universal standard, and each country or region may define their own pattern. For this reason, developed solutions have applicability in specific locality ([5], [6], [7]). Another approach [8] proposes to place RFID tags on all tactile paving surfaces, but that depends on regulation and it has a higher cost. Research sharing goals similar to our approach detects specific color paving and layout, and do not work with the tactile paving used in Brazil. Our approach detects directional and warning paving surfaces in three different colors used in Brazil.

[1]This work relates to a M.Sc. dissertation
[2]http://www.who.int/mediacentre/factsheets/fs282/en

There are already good solutions to detect ground obstacles [9], [10], [11]; however, they usually use ultrasonic sensors, cameras with depth sensors or stereo cameras to detect these obstacles. In our approach, the obstacle detection complements the tactile paving surfaces detection, being integrated in the same algorithm, without the need of a special hardware.

## III. Model Description

The proposed model is composed of four parts, as shown in Fig. 1. In this model the feedback is audible and the input data consists of images, signals of ultrasonic sensor, and GPS coordinates.

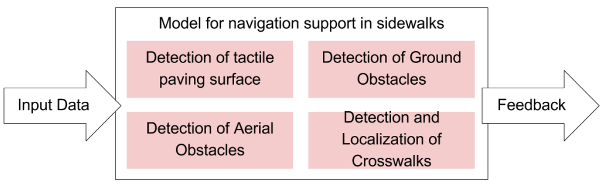



Fig. 1. Main model and its component parts.

For the detection of tactile paving surfaces and ground obstacles, a smartphone camera or webcam is used to get the images, with a resolution of 320x240 pixels or higher in landscape orientation. The camera is attached to the user's abdomen, positioned at an average height of 1.0 meter and with a $45^\circ$ angle relative to the ground.

In the detection and localization of crosswalks, the input data corresponds to GPS coordinates obtained by a smartphone and satellite, and road map images obtained through Google Maps API. Each image is a matrix with $640 \times 640$ size of resolution, with a zoom of $20\times$ (i.e., the image height from the ground), corresponding to an area of $6890m^2$, approximately.

For detecting aerial obstacles, an ultrasonic sensor was used coupled in a hat on the user's head to give information about the distance of objects. Details about each part of the proposed model are presented in the following subsections.

### A. *Tactile paving surface*

The automatic detection of tactile paving surface is an important topic of research, since it can help in the mobility of visually impaired persons, but it can also be used for the displacement of autonomous robots, providing a safe route and warnings. There are two types of tactile paving surface: guidance path or directional (Fig. 2(a)), which indicates the user a secure way, and warning surface or alert (Fig. 2(b), used to warn the user to pay more attention because of an obstacle, such as a step or entrance of cars. In Brazil, these tactile pavings appear with several color patterns, such as blue, yellow and gray.

*a) Area Detection:* This is the first step of the algorithm, in which each frame of the video is compressed into a smaller image, reduced as close as possible to 320 pixels width while maintaining its proportion. The compressed image is then converted to the YCbCr color space, or Chrominance, and a histogram for each channel (Y, Cb and Cr) is generated.

(a) Directional (b) Warning

Fig. 2. Tactile paving surfaces.

The histogram of the images presents peaks concerning the colors of the tactile paving surfaces. Those peaks combined with threshold methods are used to determine the possible tactile paving surface area in the image. Since it has different colors, we defined two configurations to represent their areas in the histogram: *Type A* which works better with brighter colors as yellow or blue; and *Type B* which works better with opaque colors as gray. This process generates a total of 6 threshold values for the 3 channels of image (Min/Max Y, Min/Max Cr and Min/Max Cb). Those thresholds, when applied to the image, can delimit the possible tactile paving surface area.

After applying the thresholds, erosion and dilation morphological operators are applied to the possible area of the tactile paving surface to eliminate false positives. The usage of *Type A* or *Type B* is determined by the lack of detection of "parallel lines" in the image. Here we considered as "parallel lines" the border lines of the tactile paving surface that point to the same direction but that are not necessarily equidistant, being able to cross at some point. The algorithm starts using *Type A* detection and if in one frame, no "parallel lines" are detected, then the last detected lines will be used. However, if the frame in which these lines were detected is 3 or more frames before the current frame, then the algorithm will change the type of detection. This feature aims to keep the tactile paving surface detection as constant as possible without allowing the user to walk on possible false positives for too long.

*b) Borders' Detection:* Since the tactile paving surface is delimited by long straight lines, one solution to detect these "parallel lines" is to use the Canny edge detector[12]. However, as the detected lines need to be smooth, and it is necessary to remove noise, a median blur filter is applied to the image before running the edge detection algorithm. Then, Hough Line Transform[12] is used on the resultant image. The results of these steps can be seen in Fig. 3.

(a) Original Image (b) Canny and Blur (c) Hough Lines

Fig. 3. Algorithms for detecting the edges.

Since the detected lines are often overlapped and represent the same line in the original image, we implemented a function that merges the detected lines that are close together and have similar angles. However, even so, there are still more than two lines in the results. Then, we chose the two most vertical lines that are also ”parallel” to each other and whose distance specifies an area compatible with a tactile paving surface.

*c) Validation Testing:* The detection of ”parallel lines” does not ensure that they would be the borders of a tactile paving surface.

Thus, the image is segmented into blocks of $25x25$ pixels that are numerated and the GLCMs (Gray Level Co-occurrence Matrix [13]) are calculated for each one of them in eight different directions.

Since the Canny edge detector is not suitable for the detection of small edges' variations, we used the Laplace operator to detect edges before applying the GLCM. Moreover, in order to improve performance, just the lower half of the image is processed. The resultant block division can be observed in Fig. 4: green blocks will be analysed and blue ones discarded.

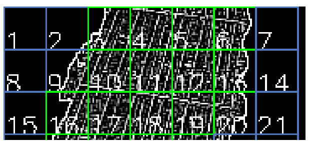



Fig. 4. Block segmentation.

For each generated GLCM, in each block, we extracted the values of entropy, contrast, homogeneity and uniformity (or energy). A spreadsheet containing 624 entries from blocks was created and, for each entry, all values were stored for each of the four directions and two distances. Using RapidMiner[3], we extracted a decision tree from these values that allow to predict the type of block: alert or directional tactile paving surface, or noise, which means everything else. In order to determine that, we used the following rules:

- **Alert:** If two neighbor blocks of up, right, left or down directions are of type Alert, then the image is considered to have an alert tactile paving surface;
- **Directional:** If the first rule does not apply and 2x2 blocks of Directional type are found, then the image is considered to have a directional tactile paving;
- **Noise:** If none of the two previous rules apply, then the image is considered as noise (without tactile paving).

### *B. Ground Obstacles*

Ground obstacle detection is another part of the model, which is fully integrated with the detection of tactile paving. It was developed because in Brazil there are few sidewalks with tactile paving. Thus, people with visual disabilities need to walk on sidewalks that often have several obstacles. In general, these obstacles are detected by them through long canes; however, the ability to anticipate danger can be a benefit for these people. In our model, any element that may hinder passage is considered an obstacle, such as holes, steps, walls and corner, as illustrated in Fig. 5.

[3]https://rapidminer.com/products/studio/

(a) Flowerbed (b) Corner (c) Wall

Fig. 5. Examples of obstacles on sidewalks.

After capturing the image of the sidewalk in front of the user, the search for the presence of directional tactile paving begins. If found, the user is guided to walk on the tactile paving. If there is a warning tactile paving, the user receives an alert sound. In the absence of tactile paving, the search for obstacles on the sidewalk starts. Texture patterns are identified on the sidewalks, and if there is a place with a divergent pattern, an alert sound is issued to the user.

We chose to work with the detection of inconsistency or change in color pattern based on the assumption that this inconsistency can mean an obstacle. With a visual analysis of sidewalk images, we detected that most obstacles on a sidewalk have a different pattern or color from the rest of the sidewalk. Fig. 6 presents a flowerbed and a car, which are examples of obstacles. This process does not detect all obstacles, but it can be processed quickly and uses the same pre-processed image for the detection of tactile paving.


Fig. 6. Image example divided into blocks.

The obstacle detection starts with the division of the image into 12 blocks, as shown in Fig. 6. Considering that the blocks closer to the user correspond to approximately $90cm$ and that an average human step has around $85cm$, this means that it is possible to predict obstacles one step ahead of the user. Block 8 is the closest to the user and it is then considered as a reference block. When the other blocks have different characteristics from the reference block, they are considered obstacles. In Fig. 6 we can see two obstacles: a flowerbed, located in blocks 1 and 2; and the back of a car located in block 9. In all these blocks, it is possible to see a different color pattern.

For each block, it is processed the average value of the channel **Cb** and **Cr**, the difference between these values, and the corresponding values of the reference block, **Diff Cb** and **Diff Cr**. Using empirically defined thresholds, the blocks that may be possibly considered obstacles are selected. To improve the processing speed, only the blocks next to the user are checked. After processing, if a block is considered as a possible obstacle, an alert tone is emitted to the user.

### C. Aerial Obstacles

The detection of aerial obstacles is the third part that makes up the model. We chose to use an ultrasonic sensor that can provide the distance of obstacles in relation to the user. The sensor is always active and gets the distance from any object that is in front of it. When the sensor detects an obstacle in a short distance from the user, an alert is emitted.

For such detection, it was necessary to set up and program equipment. A sensor was placed on a hat that the user must wear, as shown in Fig. 7. The distance information of objects that are in front of the user is sent via bluetooth to a smartphone that interprets the information and, if necessary, sends an audible feedback to the user.

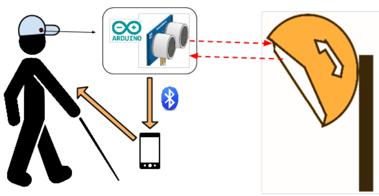


Fig. 7. Implementation of part of the model to detect aerial obstacles.

The prototype was designed so that the user can stay with hands free, with the smartphone in the pocket or attached to the body. The prototype is composed of: ultrasonic sensor (HC-SR04 model), Bluetooth module (HC-05 model), Arduino UNO R3 board, battery, smartphone and headset (optional).

Every 500ms the Arduino board captures sensor information. The smartphone receives this information via Bluetooth and if the distance is less than 1 meter, the user is notified to stop, and if the distance is between 1 and 2 meters, the user is notified to be careful. The sensor used in the prototype detects obstacles at an angle of 15 degrees. Thus, only obstacles near the user's head are detected. To detect a larger area, it is necessary to use more sensors.

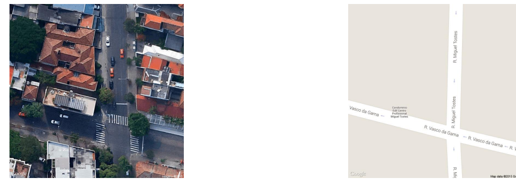
### D. Crosswalk Detection and Localization

Detection and localization of crosswalks is the fourth and last part of the proposed model. It provides the distance of crosswalks and corners from the user. This is important information to plan the navigation, in order to help the user to cross the streets more safely.

With the advent of smartphones equipped with GPS and internet access, it has become easier to obtain geographical coordinates. We use this information to get satellite and road maps of the user local images, exemplified in Fig. 8. Then, computer vision techniques are applied on these images to detect the crosswalks, as described below.

Geographic coordinates captured by the user's smartphone are sent to a Web service. This web service uses Google Maps API to get the satellite and the respective road map image, which are going to be processed.

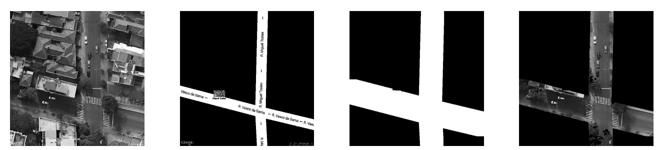
The first step of the image processing module relates to image segmentation, which consists of defining a region of interest where crosswalks could be included (on the road or near road regions). To do so, both input satellite image and road image (assigned to $\mathbf{R}$) are converted to grayscale. The region of interest $\mathbf{B}$ is generated by a simple thresholding approach. The segmentation is obtained by thresholding each pixel $(x, y)$ of $\mathbf{R}$ by $\lambda$ (where $\lambda = 245$, set experimentally). Fig. 9(b) illustrates the output of the thresholding approach.



(a) Satellite image (b) Roadmap

Fig. 8. Satellite and Roadmap images.

As we can see in Fig. 9(b), there are several undesirable structures in the binary image (text, arrows, etc). By trying to eliminate such undesirable structures, a post-processing morphological operation was applied to this binary image (a combination of erosions and dilations). The output of the morphological operations, illustrated in Fig. 9(c), was then combined with the respective grayscale satellite image (Fig. 9(d)).

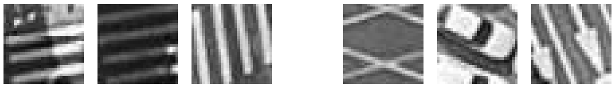


(a) (b) (c) (d)

Fig. 9. Image segmentation: (a) input satellite image converted to grayscale; (b) initial segmentation of the road map; (c) output of morphological operations; (d) combining the satellite image with the region of interest.

*a) SVM Classifier for crosswalk detection:* To detect crosswalks from satellite images we propose to train a Support Vector Machine (SVM) classifier. To this end, we built a dataset containing several positive and negative sample patches (with size equals to $30 \times 30$ pixels in each patch), used for cross-validation. This dataset is composed by 370 patches of crosswalks (positive samples) and 530 patches of non crosswalks (negative samples), illustrated in Fig. 10, which were manually extracted from Google Maps using its API (considering grayscale input images with $640 \times 640$ of resolution, captured by a zoom of $20\times$). As we can see in Fig. 10, image patches vary according to illumination conditions, and angle orientation and shapes, regarding the zebra pattern. Next, the dataset was divided into training and test set, with 600 and 300 independent patches, respectively, on which a 10-fold cross-validation was used to optimize (hyper)parameters of SVM.

We used the Local Binary Pattern (LBP) feature extraction method to train our SVM. We have also tried to use the well-known GLCM feature extraction, and a combination of both.



(a) Positive patches (b) Negative patches

Fig. 10. Manually extracted patches of positive (a) and negative (b) samples.

*b) Detection Method:* After training the SVM classifier, the next stage of the proposed model was crosswalk detection. In a practical situation, we shall consider an input satellite image ($640 \times 640$ size of resolution), delimited by the region of interest, as illustrated in Fig. 9(d). This input image is divided into small cells, with size equals to $15 \times 15$.

If a $30 \times 30$ reference patch is considered a crosswalk, a second verification is performed. In this second verification, eight neighboring patches ($30 \times 30$) are created around the reference patch, considering their neighboring cells. If at least one of these eight neighboring patches is also considered a crosswalk, the reference patch is then set as a crosswalk, otherwise the reference patch is discarded.

*c) Localization Method:* Given detected crosswalks, the next stage relates to making a spatial relation between the user and crosswalks. Firstly, we assume the user would always send his/her GPS coordinate when facing a street and, by doing so, we can easily estimate which street he/she is by analyzing the binary image (Fig. 9(c)) and his/her informed position (illustrated in Fig. 11 by a red dot). If the user is at a *corner*, we assume that he/she is facing towards the corner.

Following the estimated *street line* in both directions (left and right), as shown in Fig. 11, and also after analyzing the binary image, we can find *corner* intersections when facing another street (if that exists) and, consequently, define the *block* the user is positioned. Such information is used to discard detected crosswalks that are not connected to the *block* where the user is. *Corner* detection could also be used to give the user additional feedback.

In order to inform the user about the nearest crosswalk, if two or more crosswalks are detected, we shall compute the distance from the user to each of them, followed by the street line edges. In case two or more crosswalks have approximately the same distance in relation to the user, he/she is informed about the two nearest ones.

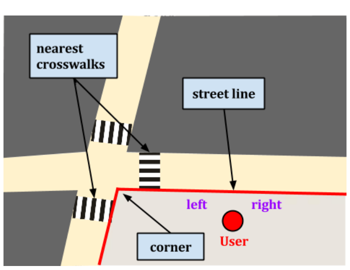



Fig. 11. Illustration of street lines, corners, blocks and crosswalks connected to the block in which the user is (information used for crosswalk localization).

Our prototype provides voice feedback to the user considering the user can be facing the corner or the street, as previously mentioned. In this way, regardless the user's position in relation to cardinal directions, feedback is provided in relation to his/her right and/or left. The provided feedback informs the distance to the nearest crosswalk (or crosswalks) on the block the user is positioned (considering only the street in front of the user and those that intersect it, which means that if the whole block is visible in the satellite image, we can discard crosswalks located on streets far away from the user).

## IV. Experimental Results

All experiments were done in a notebook with an Intel Core i5 processor, 2.27GHz and 4GB of memory. For implementation, we used C++ programming language, OpenCV library version 3.0, C#.NET 4.5 and AForge.NET framework. The images used for obstacle and tactile paving detection were obtained by a webcam model Microsoft LifeCam HD 3000.

We evaluated the tactile paving surface and crosswalk detection using a ground truth. The detection of ground obstacles was tested with volunteers, but the detection of aerial obstacles has not been evaluated for technical reasons.

*a) Tactile paving evaluation:* A total of 521 images, with different lighting settings, composes the Ground Truth: 320 are sidewalks with tactile paving surfaces and 201 are sidewalks without them. We used accuracy, sensitivity and specificity as evaluation measures, in which the specificity of 93.53% indicates that images without tactile paving surfaces are correctly predicted in most cases. We consider the specificity important for user safety, since it indicates less false positives. The sensitivity result of 85.31% and the accuracy of 88.48% demonstrate the overall efficiency of the presented approach. The processing rate was separately measured through the execution of the detection process on 7715 frame samples and the extraction of the average processing time. The obtained processing rate was approximately 16.27 fps.

*b) Crosswalk detection evaluation:* Different methods and some combinations of them were evaluated for feature extraction. First, our database with 900 image patches was randomly divided in a stratified way into a training set (with 600 samples) and an independent test set (with 300 samples). The training set was used in a 10-fold cross-validation to assess the performance of SVM for different hyper-parameters. The overall best result was obtained by LBP method, reaching about 94.6% of accuracy in crosswalk detection, with sensitivity of 95.7% and specificity of 93.9%.

For the evaluation of the crosswalk detection method, 100 satellite images were extracted from Google Maps (considering grayscale input images with $640 \times 640$ of resolution, captured by a zoom of $20\times$), taking into account the input GPS coordinate given by the user. To simulate the GPS coordinate informed by the user, we randomly selected $N = 100$ coordinate positions using Google Maps API (all coordinates were extracted from locations close to streets, simulating previously mentioned conditions in which the user is facing the street or a corner). For each extracted satellite image, a ground truth was manually generated by the user: each satellite image is defined by a binary image with crosswalk regions delimited by almost rectangular boxes. Such information is used for quantitative evaluation. Each ground truth image is confronted with the estimation given by the proposed model. The comparison is made in the level of patches instead of pixels, considering that ground truth images are discretized by patches with $30 \times 30$ of size. In this experiment, the proposed model achieved a sensitivity of 87.5%, a specificity of 97.8% and an average accuracy of 96.9%, with standard deviation of 2.841, which we consider as a satisfactory accuracy rate. The average computational cost to process each image was 497 ms, with standard deviation of 244.

For crosswalk localization we randomly chose a set of 100 geographical coordinates (simulating user's input), representing a wide number of situations, i.e., areas with crosswalk in front of the "user", on his/her right/left, as well as areas without crosswalks and areas with badly painted crosswalks or with partial occlusions. We created ground truth data associated to expected feedback for each extracted image. The proposed model provided the expected feedback in 92.7% of the simulated cases (accuracy) with an average specificity of 95% and sensitivity of 91.5%.

*c) Preliminary tests with volunteers:* We also conducted some tests with six volunteers, not blind, for detecting tactile paving and ground obstacles. Through these tests, it was possible to obtain their perceptions on the use of the prototype. Each volunteer should perform two tasks: (1) walk on a sidewalk with tactile paving, blindfolded, with only the white cane used by visually impaired persons; (2) walk on a sidewalk with tactile paving, blindfolded, with only the model prototype. Considering their answers to a survey, 83% of the volunteers agreed that the equipment provides greater sense of security, 66% agreed that running the route with the equipment has been easier than with the white cane and 83% agreed that the feedback is intuitive and enjoyable.

## V. Conclusion

In this work we proposed a new model to support navigation on sidewalks for visually impaired persons. It provides crosswalk detection and localization, tactile paving detection, and detection of aerial and ground obstacles, with low computational cost and minor user intervention. The images used for processing can be acquired by the user through his/her smartphone or camera fixed to the body, keeping the hands free. Computer vision algorithms are combined with machine learning techniques to provide information to the user.

Despite the challenges of illumination changes, occlusion, image noise and resolution, experimental results indicated that the implemented approach effectively detects the tactile paving surface, achieving about 88.48% of accuracy. Regarding the experiments with crosswalks, the results indicated that the model effectively detected crosswalks in 96.9% of cases, achieving about 92.7% of accuracy in relation to their localization. We believe that our goals have been achieved and the proposed model can be used to support visually impaired persons.

Two papers about this research have been published: one full paper was accepted for presentation and publication in 2016 International Conference on Computational Science (ICCS) proceedings [14]; another full paper is being published in the IEEE Computer Graphics and Applications journal [15]. These publications in a qualified event and journal demonstrate the quality of the developed work.

For future work we intend to use other resources available on smartphones, such as the compass and/or accelerometer, in order to increase crosswalk localization accuracy, as well as to develop a case study with visually impaired people to evaluate the real applicability of the proposed model.

## Acknowledgment

Authors would like to thank CNPq, Hewlett-Packard Brasil Ltda. and PUCRS for the financial support.

## References

[1] V. Murali and J. Coughlan, "Smartphone-based crosswalk detection and localization for visually impaired pedestrians," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE Int. Conf. on*, July 2013, pp. 1–7.

[2] D. Ahmetovic, C. Bernareggi, A. Gerino, and S. Mascetti, "Zebrarecognizer: Efficient and precise localiz. of pedestrian crossings," in *Pattern Recognition (ICPR), 22nd Int. Conf. on*, Aug 2014, pp. 2566–2571.

[3] D. Herumurti, K. Uchimura, G. Koutaki, and T. Uemura, "Urban road network extraction based on zebra crossing detection from a very high resolution rgb aerial image and dsm data," in *Signal-Image Technol. Internet-Based Systems (SITIS), Int. Conf. on*, Dec 2013, pp. 79–84.

[4] L. Shangguan, Z. Yang, Z. Zhou, X. Zheng, C. Wu, and Y. Liu, "Crossnavi: Enabling real-time crossroad navigation for the blind with commodity phones," in *Proc. of the 2014 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 787–798.

[5] X. Jie, W. Xiaochi, and F. Zhigang, "Research and implementation of blind sidewalk detection in portable eta system," in *Inform. Technol. and Applic.(IFITA), 2010 Int. Forum on*, vol. 2, July 2010, pp. 431–434.

[6] B.-S. Woo, S.-M. Yang, and K.-H. Jo, "Brick path detection from shape pattern and texture feature," in *System Integration (SII), 2011 IEEE/SICE Int. Symposium on*, Dec 2011, pp. 78–83.

[7] B.-S. Woo, S.-M. Yang, A. Vavilin, and K.-H. Jo, "Brickpath region detection using color and shape pattern information," in *Int. Forum on Strategic Technol. (IFOST)*, vol. 2, Aug 2011, pp. 720–724.

[8] M. R. B. Kumar and C. M. Sibu, "Design for visually impaired to work at industry using rfid technol." in *Electr. Design, Computer Networks Automated Verification (EDCAV), Int. Conf. on*, Jan 2015, pp. 29–33.

[9] W. C. S. S. Simoes and V. F. de Lucena, "Blind user wearable audio assistance for indoor navigation based on visual markers and ultrasonic obstacle detection," in *2016 IEEE Int. Conf. on Consumer Electronics (ICCE)*, Jan 2016, pp. 60–63.

[10] D. Aguerrevere, M. Choudhury, and A. Barreto, "Portable 3d sound / sonar navigation system for blind individuals," in *2nd LACCEI Int. Latin Amer. Caribbean Conf. Eng. Technol. Miami*, 2004, pp. 1–10.

[11] L. Johnson and C. Higgins, "A navigation aid for the blind using tactile-visual sensory substitution," in *Eng. in Med. and Biol. Soc., 2006. EMBS '06. 28th Annual Int. Conf. of the IEEE*, Aug 2006, pp. 6289–6292.

[12] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson Education, 2011. [Online]. Available: https://books.google.com.br/books?id=MaYuAAAAQBAJ

[13] A. Dixit and N. Hegde, "Image texture analysis - survey," in *Advanced Computing and Communication Technol. (ACCT), 2013 3th Int. Conf. on*, April 2013, pp. 69–76.

[14] M. C. Ghilardi, R. C. Macedo, and I. H. Manssour, "A new approach for automatic detection of tactile paving surfaces in sidewalks," *Procedia Computer Science*, vol. 80, pp. 662–672, 2016, iCCS-Int. Conf. on Computational Science, 6-8 June 2016, San Diego, California, USA.

[15] M. C. Ghilardi, J. C. S. J. Junior, and I. H. Manssour, "Crosswalk localization from low resolution satellite images to assist visually impaired people," *IEEE Computer Graphics and Applications*, vol. PP, no. 99, 2016.